DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYZING COMMUNICATION ARCHITECTURES USING COMMERCIAL OFF-THE-SHELF (COTS) MODELING AND SIMULATION TOOLS

by
Alan R. Rieffer

June 1998

Thesis Co-Advisor:	Donald Wadsworth
Thesis Co-Advisor:	John Osmundson

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1998

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Analyzing Communication Architectures using Commercial Off-the-Shelf (COTS) Modeling and Simulation Tools

5. FUNDING NUMBERS

6. AUTHOR(S)
Rieffer, Alan R.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(maximum 200 words)***

There are many initiatives focused towards the pursuit of information systems capabilities—hardware, software, and architecture—and other technologies that will markedly enhance the command and control (C2) function. The overarching purpose of this thesis is to provide joint task force communication planners with the tools for planning and managing the increasing communications demand. To this end, this project had two goals, to compare the performance of two computer-aided modeling and simulation tools representing both ends of the cost and complexity spectrum, and to provide a subjective evaluation. Four computer models were developed to simulate Information Technology for the 21st Century (IT-21) and Joint Tactical Information Distribution System (JTIDS) networks using OPNET Modeler/Radio, by MIL3, and EXTEND by Imagine That, Inc. Although assumptions were made to simplify the models, simulation runs demonstrated that the network models developed using OPNET and EXTEND produced very similar and believable results. The JTIDS models results for data rate and message latency agreed within 3.5%. Similarly, IT-21 system models detected changes and trends caused by different system loads. The results indicate that low cost, commercial off-the-shelf modeling tools can be used to describe various networks used in joint operations.

14. SUBJECT TERMS
JTIDS, Link-16, IT-21, EXTEND, OPNET, Modeling and Simulation, Command and Control, Architectures, Simulation

15. NUMBER OF PAGES
254

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFI- CATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

ANALYZING COMMUNICATION ARCHITECTURES USING COMMERCIAL OFF-THE-SHELF (COTS) MODELING AND SIMULATION TOOLS

Alan R. Rieffer
Lieutenant Commander, United States Navy
B.S., University of Arizona, 1982

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND, CONTROL, COMMUNICATIONS, COMPUTERS, AND INTELLIGENCE SYSTEMS)**

ABSTRACT

There are many initiatives focused towards the pursuit of information systems capabilities—hardware, software, and architecture—and other technologies that will markedly enhance the command and control (C2) function. The overarching purpose of this thesis is to provide joint task force communication planners with the tools for planning and managing the increasing communications demand. To this end, this project had two goals, to compare the performance of two computer-aided modeling and simulation tools representing both ends of the cost and complexity spectrum, and to provide a subjective evaluation.

Four computer models were developed to simulate Information Technology for the 21st Century (IT-21) and Joint Tactical Information Distribution System (JTIDS) networks using OPNET Modeler/Radio, by MIL3, and EXTEND by Imagine That, Inc. Although assumptions were made to simplify the models, simulation runs demonstrated that the network models developed using OPNET and EXTEND produced very similar and believable results. The JTIDS models results for data rate and message latency agreed within 3.5%. Similarly, IT-21 system models detected changes and trends caused by different system loads. The results indicate that low cost, commercial off-the-shelf modeling tools can be used to describe various networks used in joint operations.

TABLE OF CONTENTS

I. INTRODUCTION

A. TECHNOLOGY IN WARFARE: A PARADIGM SHIFT

In the last two decades computer science and data communications fields have merged with profound results. New companies combining computers and communications have emerged, producing new technology and products. The consequence of this union was a revolution in data communications. Today, computer networks seem to be growing without bound. Computer communications is an essential part of the infrastructure and commercial industry in countries around the world. Technology and technical-standards organizations are driving toward a single public system that integrates all communications and makes virtually all data and information sources around the world easily and uniformly accessible [Ref. 1]. In the United States, networks can be found at every federal, state, and local government.

The military sector has become very reliant on networks as well. The speed of communications and pace of events in the modern world have accelerated. Networks and technologies connecting workstations operated in non-combat environments, such as local area networks (LANs) with

multiple stationary nodes, are evolving to support the battlefield. At the height of Desert Storm/Desert Shield (DS/DS), the automated message information network passed nearly two million packets of information per day through network gateways [Ref. 2].

The military has been establishing a new paradigm since the Middle East conflict (DS/DS). Computer and communication technologies are being introduced at an amazing rate, the military is downsizing, and budget reductions have curtailed military spending. Civilian and military leaders are searching for the best balance between readiness and reductions. This paradigm shift is having a profound effect on operational concepts and doctrine as the services enter the 21^{st} Century. The Armed Forces of the United States face the challenge of mastering multifaceted conditions, unlike nations whose military forces can concentrate on a more limited range of environments. Forces must support an increasing number of missions, such as operations other than war, with fewer assets. The ability to project and sustain the entire range of military power over vast distances is a basic requirement to maintain stability and deterrence worldwide. This projection of power requires inter-Service linkages of modern command, control, and communications [Ref. 3].

Today warfighters rely on networks for planning, accounting, administration, logistic support and more, just as businesses in the civilian sector. When meshed with information superiority, the Joint Force Commander could deploy to the joint operations area with a smaller staff, linking back to support in theater or even in CONUS. This is particularly true if the staff function is to process and provide information rather than control immediate operations [Ref. 4].

> The nature of modern warfare demands that we fight as a team. This does not mean that all forces will be equally represented in each operation. Joint force commanders choose the capabilities they need from the air, land, sea, space, and special operations forces at their disposal. The resulting team provides joint force commanders the ability to apply overwhelming force from different dimensions and directions to shock, disrupt, and defeat opponents. Effectively integrated joint forces expose no weak points or seams to enemy action, while they rapidly and efficiently find and attack enemy weak points. Joint warfare is team warfare. [Ref. 3]

Joint Vision 2010 points out that the military must expand their tradition of joint victories, building on an extensive history of joint and multinational operations from as long ago as the Revolutionary War. Today, joint action is becoming practiced and routine. Whether there are years to plan and rehearse, as in the case of the Normandy

invasion, months as in Operation DESERT STORM, or only a few days as in Operation URGENT FURY, the Armed Forces of the United States must always be ready to operate in smoothly functioning joint teams [Ref. 3]. To this end, technologies enabling rapid information processing will revolutionize training. The 2010 warrior could have initial or refresher training available on demand. Perhaps three-dimensional (3-D) multi-sensory virtual environment mission-rehearsal training could be available on short notice. Technologies supporting this concept could include wide-band terabyte data-transfer and data-processing capability, virtual reality immersion, and fully interactive training systems. With these technologies, near-real-time information can be rapidly processed, analyzed, and assimilated for the warrior on the front line as well as the decision-maker. [Ref. 4]

> By 2010, we should be able to change how we conduct the most intense joint operations. Instead of relying on massed forces and sequential operations, we will achieve massed effects in other ways. Information superiority and advances in technology will enable us to achieve the desired effects through the tailored application of joint combat power. Higher lethality weapons will allow us to conduct attacks concurrently that formerly required massed assets. [Ref. 5]

There are many initiatives focused towards continued improvement of the Joint Force Commander's (JFC) ability to

rapidly constitute and employ the Joint Task Force (JTF). Foremost are the pursuit of information systems capabilities—hardware, software, and architecture—and other technologies that will markedly enhance the command and control (C2) function. This initiative considers the need for a common architecture and seamless interoperability among a joint force's components. This is especially important during design and procurement of information systems since information systems that are "born joint" will greatly facilitate joint interoperability. [Ref. 4]

B. MASTERING THE COMPLEXITY OF COMMAND AND CONTROL

According to Navy Copernicus, existing command, control, communications, computers, and intelligence (C4I) systems have grown to the point where there is no longer a cohesive architecture. The current Command and Control (C2) cannot support the revolution in modern war. The Copernicus Architecture outlines four knots that bind the potential power of Naval C4I. These "knots" are also applicable to Joint Operations where a robust, distributed C4I network is the key to tailored application of joint combat power. [Ref. 6]

First, there is no system or accepted technique to decant critical operational traffic from less critical or even administrative traffic. More than 33,000 commands ashore can send messages to the commander at sea. The result is that communications are driving operations, not vice versa. [Ref. 6]

Second, once the critical operational traffic is segregated, the traffic is often in the wrong format (a multiplicity of different types of narrative messages) and in the wrong form (paper). The result is the tactical commander cannot assimilate the information rapidly. [Ref. 6]

Third, there is no effective oversight of the C4I architecture. Operationally, many organizations tend to see themselves each as the "center of the universe" with the result that a host of separate communications nets, sensor formats, computer protocols, and agendas have given warfighters a much under-leveraged C4I infrastructure. [Ref. 6]

Finally, there is a loss of operational perspective. Because these critical problems are shrouded in technology, legacy systems, and C2 "plans," the true functions of modern command and control are often lost in all the fanfare of the technical solutions.

> For it is command and control, not communications and computers, nor intelligence, that is at the heart of maritime, military and joint operations. [Ref. 6]

Perhaps the most important lesson from the history of warfare is that better technology does not always prevail, instead, it is the commander that uses technology better. War does not necessarily favor the force with the most men and weapons or the side with the latest technology. It is when these elements are incorporated in a sound manner that one side gains an advantage over the opponent. Command and control systems are the tools in modern warfare whereby commanders will achieve concentration of forces.

> Joint forces now operate within a Global Information Environment (GIE). GIE is the worldwide network of information sources, archives, consumers, and architectures that provides the framework for a new global setting. The GIE is made up of many participants such as US, UN, and foreign governments; various media including a growing web of independent, on-line sources; academic institutions; a multitude of non-governmental organizations, and private volunteer organizations; complex national and international business conglomerates; and others not necessarily affiliated with any organized group. The participants operate with varying levels of independence or interdependency but all are becoming increasingly interactive in the GIE. [Ref. 4]

The Global Information Environment (GIE) may be a step toward untying the four knots that bind potential power of

C4I as described in the Copernicus Architecture. Within the GIE are complex and interconnected information infrastructures that link individuals and organizations to an ever-increasing abundance of information which provides an unprecedented interconnectivity across national lines, over Service boundaries, and between military commanders and their supporting activities. This web extends across geographic and political boundaries and presents many new unexpected opportunities as well as unique and unprecedented challenges [Ref. 4].

> …a technological development needs to have a corresponding tactical development or it becomes an engineering curiosity. Operationally it is a force divider. [Ref. 6]

There is much emphasis on stability operations, and on crises that can occur in one or more regions with little or no warning. U.S. commanders will need flexibility and combat power in the future for these scenarios. Global C4I battle management will be a prerequisite in operations other than war (OOTW). U.S. forces must be able to control the battle space wherever they operate. For effective power projection operations, the nation will be required to maintain upgraded command and control systems as force

multipliers to manage the tactical situation in joint and combined operations. Forces harnessing the capabilities potentially available from the command and control network will gain dominant battlespace awareness. [Ref. 5]

> The combination of these technology trends will provide an order of magnitude improvement in lethality. Commanders will be able to attack targets successfully with fewer platforms and less ordnance while achieving objectives more rapidly and with reduced risk. Individual warfighters will be empowered, as never before, with an array of detection, targeting, and communications equipment that will greatly magnify the power of small units. [Ref. 5]

In the vision described by Joint Vision 2010, future warfighting embodies the advances in command and control available in the information age from which, in effect, four new operational concepts have emerged (Figure 1-1): dominant maneuver, precision engagement, full dimensional protection, and focused logistics. The basis of these concepts is found in command, control, and intelligence assured by information superiority. [Ref. 5]

The bottom line is the U.S. Armed Forces depend on technological advances and use of information in support of the four operational concepts to get major qualitative advantages over potential adversaries (Table 1-1). There must be a systematic process to exploit the great potential

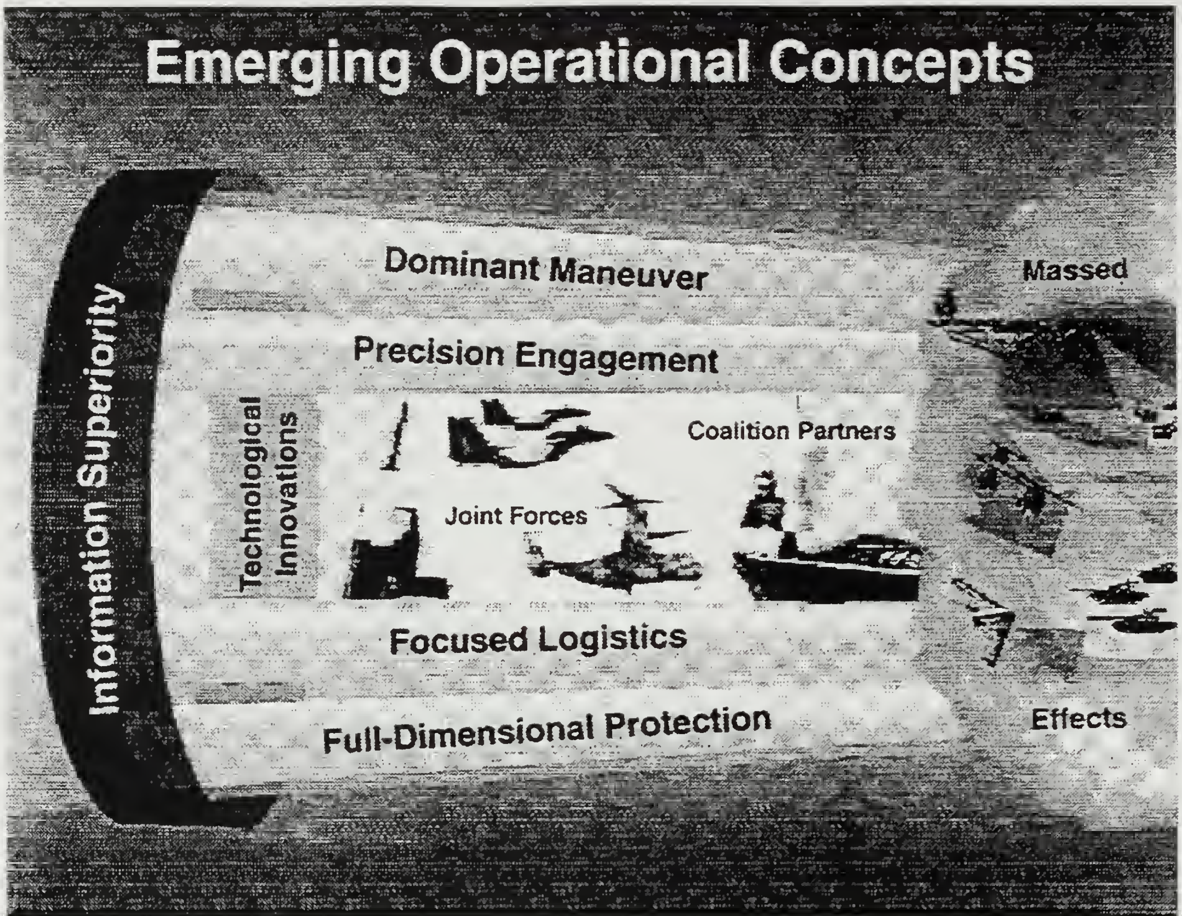

Figure 1-1. Emerging Operational Concepts From Ref. [6].

that technology can offer to command and control. Computer and communication networking is a complex subject. We can see that networks can consist of many systems forced to inter-operate and provide the necessary connectivity, data storage, and retrieval. These physical systems, in their complex arrangement, form the tangible part of command and control. Their complexity must be managed to take advantage of the asymmetry in C2 gained through technology. One of

the goals in the C4I community is to design in interoperability or "jointness" in communication systems in order to reduce the number of stovepipe or legacy systems. Meanwhile, as new technologies are introduced, the complexity and combinations of networks will continue to grow. The nature of future military operations relies heavily on mastering the myriad of technologies that make up the command and control system. To master the complexity, the services need to look beyond the stove pipe and legacy systems and understand how the C2 "system," with all of its components, performs as a whole to support the warfighter.

Table 1-1. JWCO Support for JV 2010 From Ref. [5].

Joint Warfighting Capability Objectives	*JV 2010* Operational Concepts			
	Dominant Maneuver	Precision Engagement	Full-Dimensional Protection	Focused Logistics
1. Information Superiority	●	●	●	●
2. Precision Force	o	●	o	
3. Combat Identification	o	●	●	
4. Joint Theater Missile Defense		●	●	
5. Military Operations in Urban Terrain	●	o	●	
6. Joint Readiness and Logistics	●	o	o	●
7. Joint Countermine	●		●	o
8. Electronic Combat	●	●	o	
9. Chem/Bio Warfare Defense and Protection	●	o	●	o
10. Counter Weapons of Mass Destruction		●	●	

● Strong Support o Moderate Support

The Joint Task Force Commander depends on a command and control network being in place regardless of the environment. To accomplish this task, those responsible for command and control systems need sophisticated tools to keep pace with the complexity inherent in communication networks and systems. Planners must be able to conduct short notice crisis planning and have the capability to determine, in a dynamic situation, the best way to reallocate network assets degraded due to loss or damage. This paper focuses on the use of computer aided modeling and simulation tools as decision aids for communication planners. More specifically, the target users of these tools are the communication staffs in a Joint Task Force or Unified Command. There are many situations to employ these tools, in this study the setting will be a crisis or conflict where Operation Plans or Contingency Plans can not provide the guidance needed from a command and control network perspective. Communication planners could be faced with establishing a military network compatible with the equipment, infrastructure, and operation in a matter of weeks or days. Once in a conflict, communication units will be thrust into a reactive mode, contending with equipment failures or losses due to hostile action.

This paper addresses the feasibility or utility of employing computer aided modeling tools, in a communications element or support staff, to manage the increasingly complex, heterogeneous, communication networks required to support joint and combined operations in a reactive mode. This implies a state of crisis or conflict in which the situation has gone beyond the "deliberate plan." This process of managing communications systems in a reactive mode will be referred to herein as "adaptive communication planning."

C. MODELING AND SIMULATION

1. What are Models and Simulations

In this project, "model" refers to a logical description of a system's operation or performance. Some models describe very specific operations within a system while others might describe an entire system. The amount of detail or "granularity" of the model can also vary. As can be expected, models become more complex as they describe a particular system's performance in more detail.

There are several different types of models. Mathematical models describe a system through a balance of

flow or processes represented by equations. Paper models can graphically represent functions, processes, and the relationship between them, with a system of symbols, lines, and arrows. Computer based modeling provides the same type of tracking or calculating but obviously at a much greater speed. With computer modeling software, users might be able to input several orders of magnitude more inputs or parameters which facilitates building more complex models and simulations and running more of them.

Simulation is the process of modeling the target system over a period of time or through a series of events then carrying out "experiments" to determine how the system performs or reacts. In this context, a simulation provides a means to interact with the model, which corresponds to certain aspects of the real system. There are different types of simulations as well. Two classical categories of simulations are "discrete event" and "continuous (process)." [Ref. 7]

In a discrete event simulation (simulations using discrete event models), the system or model entities change state when discrete events occur. Events are specific occurrences such as a message being transmitted, a database query, or a router receiving an encapsulated data packet. This definition of a discrete event is different than is

commonly found in engineering where the term "discrete" refers to periodic or constant time intervals. When discrete is used to describe constant time steps then it refers to a continuous simulation or model and does not have the same meaning as "discrete event" models. In this paper "discrete event" will refer to models or simulations that describe the state of a system as individual and unique entities or items occur. [Ref. 7]

Continuous simulation describes the state of a system as a function of time. The models, which these simulations are executing, are referred to as "continuous" or "process" models where states change with time. Continuous simulations are used when there is a flow of homogeneous values and time advances uniformly from step to step. Values that reflect the state of the system or model change accordingly as the time changes. For example, transmitting from a large pool or queue of messages could be modeled as a continuous system. Assuming the transmitter is sending out data at a fixed rate then the state of the system, message queue size in this example, changes with respect to time. This is a simple example and other factors such as net or satellite access time and message size and generation rate are all factors affecting the queue size.

When a system is modeled using one specific modeling or simulation tool it will be referred to as a homogeneous model in this paper. That does not preclude more than one system being described within a single model. It is entirely possible to create a model of a particular system or process with one tool, running the simulation, compiling the results, then using those results as parameters or inputs to a second model, developed with a different modeling and simulation tool. Models utilizing two or more different modeling and simulation tools will be referred to as heterogeneous models in this paper.

2. Why Develop Models

> Joint force 2010 must have a well-developed, integrated, and seamless decision-making architecture. It should leverage emerging capabilities such as artificial intelligence and micro technologies to support more efficient information fusion and multimedia, multifunctional processors capable of near real-time decision support; data compression technologies to increase speed and efficiency… [Ref. 4]

Models were defined as logical descriptions of a system's performance. It is rather obvious that models provide a means to mimic or simulate the way a system performs. The functions that communication and computer networks perform are especially well suited for computer

aided modeling and simulation. The primary concern surrounding modeling then becomes a question of utility. Can we not just simply observe and record the characteristics of the "real" system? Why expend resources to develop models or run simulations? The answers to these questions may also seem obvious but they are worth consideration to more fully understand the benefits of modeling and simulation as planning tools, especially as military information networks built from commercial off-the-shelf (COTS) hardware become more commonplace.

Perhaps the most apparent application of models and simulations is to study a system in a situation where the real system can not be used to generate the required data. This might be the case when a system is in use and conditions can not be established or the consequences of testing, such as disconnecting or shutting down to connect test equipment, are not acceptable. Another example that it is not prudent to use the operational system is when it would expose the system (which may include hardware, software, and people) to extreme conditions or inputs capable of damaging the system (or operators) or degrading performance. Most commanders are not willing to conduct this type of testing when it involves their operational systems even though the critical nature of the information

traveling through the network requires that the performance characteristics be known.

Modeling and Simulations can be a critical element during new program development and acquisition. Models can be used to justify programs, plan for future growth, and analyze reliability. C4 systems are required to provide robust communications, be interoperable, and meet desirable logistic characteristics. A C4 system is made up of four building blocks: terminal devices, transmission media, switches, control and management [Ref. 2]. The building blocks that are well suited to modeling and simulation are the transmission medium, switches, and three of the control and management functions. The transmission media is the physical path or conduit that carries the signal between terminals and switches direct information through the network to the final destination. The control and management functions that lend themselves to modeling are network performance analysis, fault isolation, and network planning and engineering [Ref. 8].

This paper investigates the utility of using computer-aided models and simulations, motivated by the concept of integrating network models with crisis action planning and real time (reactive) decision making. Consider modeling and monitoring military communication and information networks

supporting joint or combined operations from build up through conflict resolution. The answers to "Why model?" become more specific. Communication planners, with the proper tools, can quickly construct a network model representing the equipment and infrastructure available for the operation. Once the base model is built, planners can troubleshoot problems or "what if" different scenarios. Performance can be analyzed to find weak links, choke points, or identify what equipment is underutilized. If a component fails, perhaps from hostile action, alternatives can be quickly evaluated and corrective action taken. There is no "single" system or set of equipment that planners can count on for all situations. Instead they must be able to react to adverse conditions and to understand system limitations if forced to operate with shortfalls.

This generates a few new concerns about modeling. Can models provide the fidelity and accuracy needed to add any value to the decision process? What resources will the staff need to use the modeling tools? Is modeling timely? What special training will the staff need to build and use any of these tools? How should models be tested and evaluated? Answering these questions will help answer the overarching question concerning the utility of modeling and

simulation tools to support reactive or adaptive communications planning.

> With clear hindsight we can see that we have entered a new era. But only with veiled foresight are we discovering the wide range of new opportunities, seemingly endless possibilities, and significant vulnerabilities that it provides. Information Age technologies are revolutionizing the ability to collect, process, and disseminate information, and to develop the battlespace capability to "know yourself, and know your enemy" as never before. In the process, these revolutionary and previously unachievable capabilities are forcing us away from traditional notions about command, organizational design, and perhaps even the conduct of operations. [Ref. 4]

D. PROBLEM STATEMENT

Information age technologies are having a profound impact across the spectrum of military operations. The systems that provide the means for dominate battlefield awareness and complexities of the technologies that support them are increasing in a revolutionary fashion. Communication planners need decision aids and tools capable of planning, managing, and maintaining these complex communications networks which are critical to future joint operations.

1. Purpose and Goals

The overarching purpose of this research is to provide unified command and joint task force communication planners with the best tools for planning and managing the increasing communications demand. To this end, this project is conducted with two goals in mind. The first is to compare the performance of two computer aided modeling and simulation tools. The second goal is to provide a subjective evaluation to address the utility of using these modeling tools in an operational environment such as in a crisis action team or similar scenario where communicators have to be responsive to non-standard situations.

2. Problem and Assumptions

This paper compares two computer based modeling and simulations tools from relative extreme ends of the cost and performance spectrum. The two tools are Optimized Network Engineering Tools (OPNET) Radio/Modeler by Modeling Technologies for the Third Millennium (MIL3) and EXTEND by Imagine That Inc. Two communications architectures will be modeled (see Chapter II). The values generated by the models will be used to compare the tool's performance. These models will be designed to predict End-to-End (ETE) latency, effective utilization, and message buffer (queue) size.

OPNET is an established communication network modeling tool that historically has provided acceptable deterministic results. In lieu of real data, the results generated by the OPNET models will be used as a baseline.

The results will be presented in two parts; an objective section comparing the model predictions generated by each tool and second section with a subjective comparison of the two tools based on my experiences during the project.

> With clear hindsight we can see that we have entered a new era. But only with veiled foresight are we discovering the wide range of new opportunities, seemingly endless possibilities, and significant vulnerabilities that it provides. Information Age technologies are revolutionizing the ability to collect, process, and disseminate information, and to develop the battlespace capability to "know yourself, and know your enemy" as never before. In the process, these revolutionary and previously unachievable capabilities are forcing us away from traditional notions about command, organizational design, and perhaps even the conduct of operations. [Ref. 4]

E. SCOPE

The perspective for this study is crisis action planning at a unified command staff or joint task force staff level but the results can be applied to deliberate planning or lower echelons as well. The intent is to provide the communication planner with information regarding computer aided modeling as it applies to planning and

maintaining tactical military networks. Several modeling tools will be discussed but only two modeling tools will be used to develop models. This paper does not provide an all-inclusive answer to the modeling and simulation frenzy nor is this an endorsement of any of the products used or discussed. It does provide basic, but plausible, applications of models, an outline of the modeling process, and background so the operator can make an educated decision about integrating computer aided network modeling tools into his or her toolkit.

Interactive simulations used for real time training such as flight simulators are not addressed in this paper. Models, as discussed in this project, refer to those built with the specified modeling and simulation tools for specific communication systems and their corresponding simulations have been executed, with no operator in the loop, to observe performance of the specified modeling tools.

The study will include model development for specific communications architectures to assess the modeling tools. To keep the models to a manageable size, the scenarios and command and control networks modeled may be simplified or a segment identified and bounded before analyzing with the modeling and simulation tools.

Two COTS modeling tools, EXTEND and OPNET Radio/Modeler, are employed. These applications represent the low and high ends, respectively, of cost, complexity, and granularity (detail). Only two modeling tools were used to keep the project more manageable. Two communication networks or systems are modeled, Link-16 or Joint Tactical Information Distribution System (JTIDS) and Information Technology for the 21st Century (IT-21). A third model, based on a hypothetical network for near real time friendly force reporting, was dropped due to time constraints. In each case, the entire communication architecture does not need to be modeled to evaluate the effectiveness of the tools and the process. Each model will be developed using one modeling tool then compared with the corresponding model developed with the second tool. Several other modeling tools not use in this research, such as COMNET III and BEES, will be discussed in Chapter II.

F. OVERVIEW OF OTHER CHAPTERS

Chapter II provides an overview of the steps or methodology followed during this project. The chapter also includes a review of several computer based modeling tools available. The review provides a brief description of the

tools to help familiarize the reader with the software products.

Chapter III, System Architectures, covers the physical architectures of the communication networks modeled. Here you find the system descriptions, system boundaries, and assumptions of the Link-16 and asynchronous transfer mode (ATM) networks.

Chapter IV, Modeling and Simulation Tools, provides a more detailed description of the two modeling tools, OPNET Radio/Modeler and EXTEND, than was provided in Chapter II. This chapter explains the various levels or building blocks and processes employed by the tools. The descriptions should give the reader an overview of the steps or hierarchy, within each tool, necessary to understand a functional model. It is through these different hierarchies or domains that the user interfaces with the models.

Chapter V, Models, describes the logical models as built with each of the tools. Here the initial architectures, or physical models, are contrasted with the logical models to highlight differences and assumptions made in the modeling process or limitations in the tools. In some cases the models varied from the physical architectures to simplify model development and not because the tool had a limitation to emulate a certain attribute.

Chapter VI, Analysis of Results, recaps the problem statement and discusses the parameters selected for the objective performance evaluation. Some model parameters are included here that were not provided in Chapter V. The graphs, of the data collected for the analysis, are included in this chapter. Perhaps the most important section documents some of the difficulties and shortfalls experienced during this project.

Chapter VII, Conclusion, summarizes the analysis from Chapter VI. The remarks recap the trials, troubles, successes, and recommendations from the writer. These remarks represent the author's opinions, based on the experiences gained during this project. There is also a short synopsis suggesting possible future studies including military applications to refine the work started here.

II. METHODOLOGY

This project was conducted in six general phases: modeling tool selection, network definition, logical model development, network simulations, analysis of data, and developing conclusions. These phases, outlined in the first section below, describe the methodology followed during thesis development.

The second section summarizes process and factors considered when selecting the automated modeling and simulation tools to use in the project.

The last section in this chapter lists several automated tools along with a brief description of each. The purpose of the modeling tool review is to familiarize the reader with some of the products and the variety of services offered by automated software.

A. THE PLAN

Each phase appears in the approximate order it was conducted, however it was advantageous to work multiple areas in parallel whenever possible. For example, logical model development is sequenced before analysis of data in

the methodology. As it turns out, these two stages were not independent, sequential steps. Data probes to support the analysis phase were required to complete the logical models. This generated a need for at least a draft analysis plan to identify the data collection requirements, which in turn defines the data probes requirements, and the probes could be built into the models to extract the information. In addition, there were phases that went through several iterations before arriving at the final product.

1. Select Modeling And Simulation Tools

- Modeling Tools shall have Discrete Event and Continuous (Process) Modeling Capability
- Select One Tool Based on Capability to Support Detailed (High Granularity) Modeling (System Resources and Ease of Use not an Issue)
- Select One Tool Based on Price (low), Apparent Ease of Use that can run on a Personal Computer (PC)
- High Granularity (detail of model) is not required for the Lower Cost Modeling Tool
- The Low Cost Tools Shall be COTS
- Tools Available at Naval Postgraduate School
- Select Two Computer Aided (Automated) Modeling Tools

2. Define Networks (Physical Models)

- Identify Communication Systems
  - Select Two Systems
  - Systems Must have Network Function
  - One System will be a legacy system
  - One System Shall be a Military System
  - One System will be based on COTS Equipment
  - Both Systems should have Joint Applications
  - Select two Functionally Different Systems
  - Systems will Support Voice and Data
- Define Physical Architectures (Networks)
  - Identify Granularity (System Level)
  - Establish Physical Bounds or Limits to Systems
- Determine System Test Configuration and Lineup
  - Establish System Mode of Operation
  - Identify Network Protocols (if applicable)
  - Select Operating parameters (Bandwidth, Frequency, Data Rate, Network Load)
- Determine Control Test Parameters (Network Loading)
- Determine Measures of Performance
- Record Assumptions and Simplifications

3. Develop Models (Interactive Process)

- Paper Model of Links, Nodes, Interfaces
- Identify Bounds of Logical Model (Size and Detail)
- Build Network Model
- Verify Model Consistent with Analysis Plan
- Build Network Nodes and Data Links
- Build Process Models (as required)
- Incorporate Data Probes (OPNET) or Plots (EXTEND)
  - Probes and Plots for System Troubleshooting
  - Sensors to Collect Performance Measures
- Test and Refine Model
  - Does the Model Generate the Required Data
  - Test with Pre-Determined Control Settings

4. Run Simulation (Iterative Process)

- Test Model
  - Run Simulation with Control Data and Parameters
  - Refine Model as Required
- Run Simulations
  - Set Desired Network Load
  - Perform Required Runs
  - Record Test Parameters
- Collect Data

- Record Output Scalar and Vector Files (OPNET)
- Record Plots and Data Files (EXTEND)

5. Analyze Results

- Review/Develop Problem Statement
- Determine Type of Analysis (e.g. Pair-wise Comparison)
- Define Measures of Performance
- Identify Data Required for Analysis
- Determine Number of Data Sets/Runs
- Compare Results between Modeling Tools
- Compare Model Results with Live Data (if available)

6. Draw Conclusions

- Evaluate Statistical Results obtained from Simulations
- Make Objective Conclusion Based on Performance Measures
- Make Subjective Conclusion Based on Experience with Both Models
  - User Friendly
  - Online Help
  - Documentation

- Utility of Using Tool in the Context of Crisis Management and Planning

- Suggested Future Studies

B. SELECTING THE AUTOMATED TOOLS

The number of computer based modeling and simulation tools available today is staggering. There always seems to be a newer, bigger, or better tool for the job. This is just the nature of the technology. This project makes the assumption that there are robust automated tools designed specifically for network modeling that can simulate the performance of a communications network at very detailed levels or high granularity. Another assumption is that communication planners in a crisis action team or similar situation need a tool that provides a good approximation of overall system performance, not how many bits and packets are lost or collided. More robust support is available at rear echelons if needed. This study is concerned with finding a tool with the flexibility to model a variety of communication networks and the ability to approximate system performance at a macro level such as system throughput.

Several characteristics, in addition to performance, were considered when selecting the modeling tools for this

project. This section provides a comparison of tool attributes and desired characteristics. Chapter IV contains a detailed description of the two tools selected.

The tools selected for the project were "OPNET Radio/Modeler," version 3.5, by Modeling and Technologies for the Third Millennium (MIL3), and "EXTEND," version three, Performance Modeling for Decision Support, by Imagine That! Incorporated. Both tools have a discrete event and continuous (process) modeling capability.

OPNET was developed specifically for communication and network modeling. It has several pre-built processes, nodes, and networks that can emulate computer networks and the effects of radio propagation. As such, OPNET is the tool selected to support detailed (high granularity) models.

The next characteristics considered were cost (low), ease of use, and the ability to run on a PC. EXTEND has all of these attributes. The scientific versions of EXTEND cost about $700 places it in the lower end of the cost spectrum. [Ref. 9] compared to about $15,000 for OPNET Modeler Radio. "Ease of use" can be interpreted in many ways. In this context it refers to being user friendly, not requiring special equipment, and portability of necessary documentation (users manuals). User friendly is a very subjective attribute. Past experience with this tool and

EXTEND's simple, graphical users interface made it easy to build functional models. The online help and the EXTEND users manual (one paperback book) filled in the detail on using the pre-built objects. Both EXTEND and OPNET have versions that can be run on a PC. EXTEND can also run on a Macintosh, which adds for flexibility.

The ability to model a communication system or computer network using pre-built objects or code was not a requirement of the lower cost model. The tool did need to have queues, timers, event generators, random number generators, variety of distribution functions, math function, and the ability for the users to build their own objects without knowing the program language. So in this sense, EXTEND did not have pre-built objects to build detailed models of communication networks but it would allow the user to create the objects necessary.

Both EXTEND and OPNET are COTS products. The lower cost tool had to be a COTS product primarily to be consistent with the trend of going away from legacy systems and government developed systems [Ref. 5].

The tools had to be available at the Naval Postgraduate School (NPS) simply because that is where the majority of the research was taking place. This was not a factor in selecting OPNET since there were other high fidelity

modeling tools available at NPS Monterey. EXTEND was also available at NPS and that influenced the decision to use that tool in the research but EXTEND also had all the desired attributes.

Only two modeling tools were used to develop models and simulations for this project. This was necessary to keep the scope of the project manageable. However, the tools selected represent relative extremes of cost and complexity. The results from developing models and running simulations should bound the results obtained from using most of the other modeling tools available. Even building models and simulation with just two tools required the majority of the time on this project, which is attributed to learning how to use them.

C. REVIEW OF AUTOMATED MODELING AND SIMULATION TOOLS

There are numerous modeling and simulation tools available. The information collected here is a starting point for organizations that are interested in developing a communication or network modeling capability. The automated tools discussed here only scratch the surface. Those that are covered have some capability to model communication systems and networks. This is primarily a compilation of

reports or assessments performed by outside agencies and not the author's evaluation of the products. The level of detail and format vary between products simply because the information was extracted from several different sources. The descriptions do outline some of the characteristics to consider when deciding whether or not to add computer aided modeling to your toolkit and which tool to select. The intent is to introduce different products and present observations and evaluations of automated tools. Reference 10, Air Force C4 Agency (AFC4A) Technical Report, is a sample evaluation of several automated modeling tools. The 1995 report is somewhat dated but the results and the measures are worth reviewing.

1. Battle Force EMI Evaluation System (BEES)

BEES is a large-scale modeling and simulation tool that is in development by the SPAWAR Systems Center under the sponsorship of the Joint Spectrum Center, Plans and Programs Directorate (J5). BEES provides interactive simulation of up to 2000 platforms conducting warfare operations with up to 64 systems on each. The resulted generated by the electronic battlefield are used to simulate the performance of systems in a dynamic electromagnetic environment (EME). The "simulation software," which runs the scenarios,

provides the user with a windows based Motif graphical user interface (point-and-click). The user interacts with the "analysis package" through Dialog Boxes to extracts and displays data during and following a simulation. BEES also provides a comprehensive database, constructed using object-oriented design (so was the simulation software). Selecting a ship by name or hull incorporates all the ship's systems and associated parameters. Platform characteristics and parametric data are available for over 20 types of platforms including aircraft, chaff, ships-submarines classes, radar and electronic support measures (ESM), navigation aids, shore bases, and sonobouys.

BEES has about 25 pre-built models to simulate behavior of platforms, weapons, sensors, and communication systems. Models include but are not limited to radar, communications, frequency hopping communications, reporting, jamming, ESM, satellite, electromagnetic interference (EMI), motion and maneuver, and flight operations. Orders, such as platform movements and weapons systems employment, can be entered from prepared scripts, interactively from the keyboard during the simulation, or both. This gives BEES an interactive capability that might be useful for training or assessing different actions. Data is collected and stored in history files through out the simulation (as defined by

the user). BEES also contains at least five basic scenarios that can be used for templates.

BEES runs on a stand-alone workstation. A BEES workstation is made up of a DEC VAX VMS 3100 or 4000 workstation and VAX Storage Works to provide up to three gigabytes of removable storage. [Ref. 11]

2. COMNET III

COMNET III is available from CACI Products Inc. for $25,000 to $35,000. COMNET is a commercial off-the-shelf application written in about 150k lines of Modsim II, a language also by CACI Products Inc. The function of COMNET III is to estimate the performance characteristics of computer based networks. It was developed primarily to model Wide Area Networks (WANs) and Local Area Networks (LANs). Recommended uses are [Ref. 12]:

- Evaluating grade of service contracts
- Evaluating performance improvement options
- Introduction of new users/applications
- Network sizing at the design stage
- Peak loading studies
- Resilience and contingency planning.

COMNET III is considered a programming-free communications network simulation tool. It employs a graphical user interface to create a network description. Objects are created which represent various pieces of hardware that are found in the network. These objects make up the basic building blocks of the network. Creating representations of all the different possible equipment in a network would be unwieldy. Instead, the objects in COMNET III are built with characteristics that the user can edit to represent the specific piece of equipment being modeled. These basic building block or objects represent hardware items such as computer and communication nodes, router nodes, ATM nodes, and the links.

COMNET III can run on a PC and uses a standard Windows^{tm} interface. Model definition is quickly and easily modified, allowing for experimentation and dynamic analysis. It is designed to model a variety of network topologies and routing algorithms to include Institute for Electrical and Electronic Engineers (IEEE) 802 standard protocols such as circuit, packet, virtual, and message switching. COMNET III also has the capability to archive predefined and user-defined objects and the latest release introduces wireless modeling functionality. The report generator outputs the results after running a simulation. [Ref. 8]

3. Distributed Queue Dual Bus Simulator (DQDBsim)

DQDBsim is a beta version simulator for the Distributed Queue Dual Bus Metropolitan Area Network protocol. DQDBsim provides simulation of Queued Arbitrated (QA) service, based on the protocols described in the IEEE standard 802.6. DQDBsim provides a single-process discrete event simulation of the protocol. There may be a production version of this simulator available today. There are also other modeling tools that can perform the same functions as DQDBsim with more robust technical support. [Ref. 13]

4. EXTEND Version 3.X

EXTEND Version 3.X is developed and distributed by Imagine That! Incorporated. This is a dynamic simulation environment, which supports discrete event, continuous, and combined discrete event/continuous processes models and simulations. EXTEND comes in four basic configurations. The basic configuration provides continuous modeling, science and engineering version. Other configurations add business processing and manufacturing functions.

The EXTEND libraries contain a large selection of pre-built building blocks. No programming is necessary. The blocks are grouped according to function and represent basic

processes or actions. This makes it easier for new users to quickly grasp their functionality. Represented by icons, the blocks are easily assembled by dragging and dropping them into the working space. The user connects the blocks in the desired sequence, enters the parameters into each block through dialog pages, and the model is ready to run. The items within the dialog boxes are already defined based on the block's functionality. The user just fills in the blanks with the desired parameters or information. A more detailed description of EXTEND building blocks and model development is in Chapter IV, Modeling and Simulation.

As models grow and become more complex, the user can group these building blocks and consolidate them into higher level hierarchical blocks with all the inputs and outputs still represented on the upper level block. Users can even build their own blocks using an installed block template or by modifying an existing block. There are provisions for users to add their own remarks, notes, and titles throughout the model.

Data can be entered into the block dialogs, interactively, or read in from files while the simulation is running. After a simulation has run, dialog boxes hold vital simulation information like utilization rate, number

of items entering or leaving the block, queue length, and more.

EXTEND runs on Macintosh or windows machines. It cost about $700 for the basic modeling tools and about $1285 to get the complete package.

5. MATRIX/SYSTEM BUILD

Integrated Systems Inc. of Santa Clara, California produces MATRIX/SystemBuild. SystemBuild uses a visual design environment, which forms the core of the $MATRIX_X$ product line. First introduced in 1984, the SystemBuild environment has evolved into a graphical based tool for modeling and simulating complex dynamic systems and testing control/software algorithms. [Ref. 14]

SystemBuild models are built by grouping basic building blocks into functional units or SuperBlocks. These blocks are reusable, allowing for a hierarchical design structure and simplification of complex functional units. Levels of hierarchy are limited only by the capacity of the system to allow functional decomposition of complex systems. [Ref. 14]

SystemBuild packages user defined functional designs into a single entity, or component, that is treated like a built-in block. Components are created and managed via a component wizard. All user-defined blocks can be added to a

custom block palette and used in distributed environments, facilitating the exchange of information. [Ref. 14]

The SystemBuild simulator currently supports ten integration algorithms for high-fidelity simulation of continuous systems: Euler's Method, Second-Order Runge-Kutta, Fourth-Order Runge-Kutta, Fixed-Step Kutta-Merson, Variable-Step Kutta-Merson, Differential-Algebraic, Stiff System Solver (DASSL), Over-Determined (ODASSL), Variable-Step Adams-Moulton, and QuickSim. This wide variety of algorithms enhances simulation through control over numerical accuracy of simulation parameters. MATRIX products come in both UNIX and Window NT versions. [Ref. 14]

6. MODSIM II

MODSIM II, by CACI products Inc., is an object oriented language simulation originally developed under contract to the U.S. Army. The language compiles to C for a variety of platforms. MODSIM is based on the process-oriented view. Objects have classes with various processes that can make changes to the instances of the class. MODSIM II includes a graphical simulation animator interface to build user screens, icons, and menus. It is a concurrent programming language with mechanisms to provide for pausing and

synchronization with other objects or with the system clock. [Ref. 13 and Ref. 15]

7. Optimized Network Engineering Tools (OPNET)

OPNET, by Modeling Technologies for the Third Millennium (MIL3), is a comprehensive modeling system capable of simulating large communications networks with detailed protocol modeling and performance analysis. Some of the features include graphical model building, event-scheduled Simulation Kernel, data analysis tools, and hierarchical object-based modeling. OPNET offers a library of several pre-built models and a model building wizard for rapid model development. The program also provides the modeler with the flexibility to develop unique networks. The Radio/Modeler version supports mobile radio packet modeling (satellite orbits, user defined trajectories). OPNET's hierarchical modeling structure accommodates special problems such as distributed algorithm development. [Ref. 16]

The OPNET program is window-based, utilizing a graphical user interface (GUI) similar to those used by other interactive software applications. It uses windows, dialog boxes, buttons, and scroll bars, and point-and-click for input whenever possible. The OPNET program supports

several window systems including UNIX, Sun Open Windows, HP Visual User Environment, and Windows NT. Because OPNET is GUI-based, it cannot be used from an ASCII terminal. It can only be used from a graphics workstation console or X terminal [Ref. 16].

The information in this section is a broad overview of the OPNET system. A more detailed description of OPNET Radio/Modeler is provided in Chapter IV, Modeling and Simulation Tools.

8. Prophesy Version 2.0C

Prophesy, by Abstraction Software, is a low priced discrete event Windows-based network and workflow simulation system. For about $600 Prophesy provides a network workflow simulation system with message flow animation, a feature usually found in more expensive simulation packages. It features a graphical user interface, drag-and-drop functionality for model construction, and embedded verification, confidence analysis and costing features. Abstraction Software advertises Prophesy's easy-to-use, but powerful simulation environment, allows for rapid prototyping and concept modeling, while permitting incremental modeling of more advanced features. Prophesy runs on a 386, 486DX or Pentium, with 4 MB of RAM memory

under Microsoft Windows™ 3.1 or above [Ref. 17]. There were no Macintosh versions listed in the literature but check with the vendor for the most current information. System requirements are listed as a Personal Computer with Four Megabytes of RAM and five Megabytes of disk memory. [Ref. 13]

A demonstration of Prophesy is available at "ftp://ftp.csn.org/abstraction/prophesy.exe." The demo uses the actual Prophesy interface and walks you through a model creation, and simulation run using a pre-recorded model of a simple network. The demo, contained in file PROPHESY.EXE, is a 1.2-Megabyte self-extracting file.

This package may be a good multipurpose modeling tool to get a first order of magnitude prediction of system performance. For example, users could capture all the back-of-an-envelope calculations that become unwieldy as systems grow and become more complex.

9. Queuing Network Analysis Package 2 (QNAP2)

QNAP2 is maintained and distributed by SIMULAG, with cooperation of INRIA, and marketed in the United States by Techno Sciences Inc. (TSI) located in Greenbelt, Maryland. It was originally developed as a research tool for queuing systems scientists.

QNAP2 is an object-oriented algorithmic language capable of developing high-level, complex models with powerful analysis tools. Attributes include a set of analytical solvers implementing several different queuing theorems, a Markov chain analyzer, and a discrete event simulator. Each method (analytical, Markov, and discrete event) computes several basic performance indices: server utilization, throughput, queue length (mean, maximum, standard deviation, and distribution), service time, and response time. Results can be separated for each customer class. The simulator calculates confidence intervals and allows the user to specify performance indices. QNAP2 will run on a PC. [Ref. 13]

10. REAL

REAL is a network simulator based on the NEST simulation package developed by Columbia University. The information on REAL was sparse but it is listed here because it described as a realistic and fast simulation of transport layer protocols with specific reference to congestion control. REAL will run on SUN, Vax, and Mips machines.

11. SES/Workbench®

SES/Workbench® is a commercial off-the-shelf product developed by Scientific and Engineering Software (SES), Inc. in Austin, Texas. It is a visual simulation environment with a graphical user interface to build and execute complex models for performance analysis and functional verification. A model is developed in a hierarchy consisting of three levels: graphical, directed views or graphs; declarative, filling in forms attached to each node in a graph; and procedural, specifying procedural methods attached to the nodes using a proprietary SES language which is a superset of C. [Ref. 13]

SES/Workbench® has pre-defined building blocks for queues, management of resources, transaction flow control, concurrency and synchronization, and submodel management. The execution of a model has an animated capability to demonstrate the flow of transactions through the graphs, displaying dynamic statistics, and support trouble shooting. Other features available are model libraries, a query facility to read and write models, dynamic heterogeneous simulation, and graphical statistics processing. [Ref. 13]

SES/Workbench runs on AIX, Sun SPARC OS, Sun Solaris, HP/9000, and HP-UX systems. SES announced the availability of a Build-n-Run for Windows NT®. This tool is the first

phase of redesigning SES/Workbench® for NT and UNIX platforms. The SES/Workbench Models are first constructed on a UNIX platform then Build-n-Run enables those models to run on a Windows NT platform. SES reports they will continue to support and develop Workbench on the existing UNIX platforms. [Ref. 18]

12. SES/Strategizer®

SES/Strategizer® is an application tool to conduct performance analysis of client/server systems through simulation. SES/Strategizer® provides a graphical modeling interface for defining network topologies and characterizing the performance of conventional client/server system components such as computers, networks, interconnections, databases, and application software. SES/Strategizer based on a client/server simulation model developed with SES/Workbench®. SES/Strategizer® runs on Microsoft® Windows NT™ workstations. [Ref. 19]

13. SPECTRUM XXI

SPECTRUM XXI is a Department of Defense (DoD) spectrum management system for Joint Operations and sustaining base activities. This automated tool is considered a "best of breed" product, combining the capabilities of current

frequency management systems. This is primarily an automated management tool containing a variety of canned models describing electromagnetic emissions. The purpose of SPECTRUM XXI is to provide spectrum managers with one tool to meet the needs. In concept, it will be used from the Joint Task Force (JTF) to the Post, Camp, or Station as well as the Joint Spectrum Center. Features include spectrum management support tools (point to point analysis, skywave prediction, coverage plots, spectrum occupancy graphs), automated frequency assignment, interference analysis and reporting, automated satellite access management, electronic warfare (EW) support and an editor. Permanent and temporary frequency assignments can be archived in the SPECTRUM XXI database. SPECTRUM XXI also provides for automated distribution of spectrum management data via the secure IP Router Network (SIPRNET) or remote STU III dialup. [Ref. 20]

14. Architectures Design, Analysis and Planning Tool (ADAPT)

Architectures Design, Analysis and Planning Tool (ADAPT) was developed by Mitchell Systems under Defense Information Systems Agency (DISA) sponsorship. It is designed to automate the characterization of information systems infrastructures with graphical representation

(architecture planning). Representations emulate hardware, software, data, communications and their relationship to re-engineering initiatives. ADAPT allows multiple architectures to be queried while viewing them using a unique relationship between computer-aided design (AutoCAD) and relational database technology (Oracle). The model architectures are built using a graphical interface where users can drag-and-drop representations of objects, such as terminals and satellites, to a design palette. Users populate each object information fields through simple dialog boxes. ADAPT operates on a stand-alone personal computer with AutoCAD to generate graphics. [Ref. 10]

15. Air Force Satellite Control Network (AFSCN) Performance Simulation and Analysis Tool (APSAT)

Air Force Satellite Control Network (AFSCN) Performance Simulation and Analysis Tool (APSAT) was developed by OTI to model and simulate computers, computer networks, and the workload models used to analyze system performance. APSAT is a Microsoft (MS) Windows based front and back-end for Network II.5, a simulation language and simulation engine developed by CACI Products Inc. APSAT uses a graphical user interface to build network models and a reusable library of the model objects created. Is has an automated Network II.5

simulation code generator and presents simulation results in graphical or tabular format. Probes can be selected or defined to capture any results generated during a simulation. This includes performance indicators such as utilization of system components, throughput and user response time. APSAT operates on a stand-alone PC with Window 3.1 or better. [Ref. 10]

16. Foresight

Foresight is a UNIX based general-purpose simulation tool that uses data flow diagrams, state machines, and software code blocks to perform simulations. Models are built using a graphical user interface tools. The current release contains over 100 pre-defined library elements, including signal generators, filters, queues, and process resources (CPUs, buses, etc.). Additionally, it supports the development of user-defined reusable elements. [Ref. 21]

Foresight also supports interactive simulation. External responses are sensed using manually operable input devices in an on-going simulation. These inputs are recorded and can later by used as repeatable inputs for simulations run with different model configurations.

Foresight operates on a UNIX workstation in a client-server or a stand-alone configuration. It costs about $24,000. [Ref. 10]

17. NetViz 3.0

NetViz V2.5, by Quyen Systems, is a refined graphical tool suited to a variety of applications, including documenting computer and telecommunications networks, systems processes, and other multi-level real and conceptual structures.

NetViz comes with a collection of built-in node types that represent each device on the network, such as a workstation, printer, server, or Ethernet backbone. The user selects the objects from the node list and drops them into the diagram, populating the network based on information contained in the devices. There is also an auto-discovery feature to assist with node selection. Nodes are then connected by "links" with properties consistent with the network such as 10BaseT or Ethernet coaxial. The user can customize nodes and link types by modifying the object catalog in the network diagram.

NetViz 2.5 operates on a PC in a client-server or stand-alone configuration. It costs about $595. There is a

demonstration of the latest version 3.0 from the Quyen web site. [Ref. 22]

18. Physical Architecture Application (PA2)

Physical Architecture Application (PA2) Version 3.0 is a database management application developed by the Air Force C4 Agency using Paradox for Windows Data Base Management System. The purpose of the product is to simplify the task of capturing data useful for characterizing C4I systems. PA2 can automatically generate C4I system interface diagrams based on recorded system data. Users are can access numerous data entry/collection forms and other data management functions. Other features include utilities for data submission for distributed gathering, data merging, and reports. PA2 operates on a PC with Window 3.1 or greater in a client-server or stand-alone configuration. [Ref. 10]

19. RDD-100

RDD-100 Version 4.02, by Ascent Logic Inc., is a COTS simulation tool. It utilizes a structured executable language, which implements an entity-relationship-attribute data model. The model is implemented as an object-oriented database, manipulated by a textual interface, graphical interface, or both. The graphical user interface is used to

describe system behavior in terms of input sequences and timing, model functions, and simulation outputs. This interface also provides the user a report writer to describe the dynamic properties of systems in the terms used to prepare specifications and other documents. RDD-100 also generates static views, such as behavior diagrams and Integration Definition (IDEF) functional graphs (IDEF0 graphs). The product is an object-oriented, discrete event simulation that models the systems functional behavior.

RDD-100 is available for multiple platforms including Macintosh, DOS, Windows, Unix, and Sun. It will operate in a client-server or stand-alone configuration. Price ranges workstations from about $22,000 (Partial) to $65,000 (Full function). [Ref. 10]

20. Sterling Developer

Sterling Developer, by Sterling Software, is a COTS, PC based, computer aided software engineering (CASE) tool for system analysis, design, and planning. It provides graphics capabilities to draw; store; and reference and/or link all diagrams, matrices, and screen/report layouts generated. All diagrams have automatic drawing and routing of connectors between objects. Icons represent objects. Users can customize and create icons from a palette of shapes. The

objects, properties, and relationships are maintained within a data dictionary. The query feature lets the user retrieve select information from the repository. It also allows the user to define, alter, and reuse report and display formats. The application can limit access to any information or diagrams, at any level of abstraction. Sterling Developer maintains an audit of access information such as date, time, and authorized user for creation and last update. This application runs in a LAN environment, stand-alone, or on-line with the central repository facility. [Ref. 10]

21. System Architect

System Architect Version 4.0, by Popkin Software & Systems, is a COTS CASE tool that supports the requirements and design phases of system development life cycle. It contains a data dictionary/encyclopedia with diagramming capabilities. System Architect supports multiple structured analysis and design methodologies through graphical representation of system including data flow, structure charts, entity-relationship diagrams, IDEF0, IDEF1X, structure charts, state transition diagrams, decomposition diagrams, and flowcharts. In addition, System Architect supports an automated documentation facility, spreadsheet interface, tracking of an unlimited number of project and

corporate definitions, audibility, and reusability. Data dictionaries and encyclopedias can be merged from multiple stand-alone users.

System Architect is a, PC based, stand-alone application. It runs on Window 3.1 or better and cost about $1400. [Ref. 10]

22. Tactical Network Analysis and Planning System (TNAPS)

Tactical Network Analysis and Planning System (TNAPS) Version 1.0, by Logicon, is described as a DOS based series of programs developed for use in planning, engineering, and managing tactical communications networks in both exercise and operational scenarios. TNAPS maintains a database of information for each network defined. Operators can model tactical communication plans through a graphical user interface, then extract much of the database information from those models. Planning is conducted at two levels: network and nodal/equipment. The tool can generate pre-formatted reports completed with the planning and engineering data. TNAPS maintains a database containing very broad communications and network equipment and connectivity information. [Ref. 10]

Automated modeling and simulation tools are evolving as fast as the systems they are developed to model. That makes any evaluation of these tools a perishable product. The Air Force C4 Agency (AFC4A) 1995 Technical Report, Ref. 10, documents their evaluation of several automated modeling tools. The information is dated but the measures are worth reviewing as an approach to conduct an evaluation of tools in the reader's particular area of emphasis.

III. SYSTEM ARCHITECTURES

Two very different communications architectures were used as templates to develop models with the automated modeling tools described earlier. The intent of modeling different systems is to provide a means to compare the modeling tools and their utility in a Crisis Action Team environment. The first section in this chapter identifies the two systems and briefly explains why these networks were selected. The last two sections present a detailed description of these two communication architectures and identifies the segments modeled or processes simulated for this project.

A. SELECTING THE SYSTEM ARCHITECTURES

Joint and coalition forces have come to rely on a variety of communication systems for command and control. The different systems and components can be combined into a virtually endless number of architectures. To provide some insight on how the modeling tools can support planning by modeling just two systems it was necessary to identify two broad categories of networks. The categories identified were networks with guided transmission media and systems

using wireless transmissions. One system from each category would be modeled. There are of course hybrids or heterogeneous systems but by modeling networks based on each type of transmission media it demonstrates flexibility in the modeling tools. Once the categories were established, some rather basic system characteristics stood out as being key to selecting the systems modeled during this project.

First, the system had to be a military system or one that had an identifiable military command and control application as discussed in Joint Vision 2010 or Concept for Future Joint Operations. The list was still quite large but now the focus turned toward systems that might be a factor in operations other than war (OOTW), precision strike or perhaps light intensity conflict.

Next, the baseline systems must be data networks or have a data networking capability. This ruled out single purpose, point-to-point, voice radio communication systems. This characteristic may seem obvious but is important to be consistent with the stated scenario of using computer aided modeling tools in a crisis situation to help planners manage command and control networks.

The third characteristic came from the desire to have some contrast between the systems selected and therefore provide insight into the diversity of the modeling tools

employed. This roughly interprets to identifying two system architectures that differ in the way that the data is packaged, handled, communication medium, multiplexing techniques, or even the way the network is managed.

Finally, it was important to find a fielded system with joint applications and an established military entity interested in measuring or predicting one or more aspects of system performance with a computer based model. The purpose of selecting a fielded system is you get along with it system experts, a well-defined architecture, and possibly real world performance data to validate the computer model. An established military entity can help provide the resources necessary for researching the communication system and developing the models.

In the end, Link-16 or Tactical Digital Information Link (TADIL) J, and Information Technology for the 21st Century (IT-21) were selected as the baseline network communications systems for modeling. The two systems share characteristics with many of the communication systems used by the military today. Link-16 uses Time Division Multiple Access (TDMA) multiplexing as do other situational awareness (SA) systems such as Situational Awareness Beacon with Reply (SABER) and Enhance Position Location Reporting System (EPLRS) [Ref. 23]. IT-21 uses asynchronous transfer mode

(ATM) technology, which is a high speed, flexible protocol used with Broadband Integrated Services Digital Network (B-ISDN) and many non-military applications.

B. LINK-16

The U.S. Navy uses the North American Treaty Organization (NATO) designation Link-16 when referring to Tactical Digital Information Link (TADIL) J. The U.S. Joint Services other than the U.S. Navy employ the latter term. Link-16 combines TDMA, frequency hopping, and direct sequence spread spectrum technologies in a UHF radio network for real time exchange of tactical data. It is planned for the backbone of the Joint Tactical Information Distribution System (JTIDS).

The general purpose of Link-16 is the same as the legacy systems Link-11 and Link-4A. That is to provide the exchange of real-time tactical data among units in the force. Link-16 introduces several new characteristics that the previous data links lacked. It is considered a node-less architecture with improved jam resistance, flexibility of operations, separate data and transmission security, provisions for more participants, increased data rate (capacity), and a secure voice feature. Link-16 also

provides two layers of communications security, message security and transmission security. Message security is related to message encryption. Transmission security relates to system jitter, a 32 bit pseudo-random noise variable, and the frequency hopping pattern of the carrier. System jitter and frequency hopping pattern are discussed below.

Link-16 uses Time Division Multiple Access (TDMA) to form virtual channels using the same radio frequency spectrum. In TDMA networks, information or data is broken into small, predetermined fixed size packets. Each packet is transmitted at a specific time and in a specified fixed length window or Time Slot which makes Link-16 a synchronous system. Link-16 Time Slots are 7.8125 msec duration and uniquely identified by their sequence within the overall TDMA cycle defined as an "Epoch." An Epoch is 12.8 minutes and consists of 98,304 Time Slots. The Time Slots are separated into three interleaved groups called "Sets," designated A, B, and C with 32,768 Time Slots each. The Sets are interleaved so there are two time slots from other sets between two consecutive time slots in the same Set. For example, the first six Time Slots in an Epoch are A-0, B-0, C-0, A-1, B-1, and C-1. The number indicating the Time Slot sequence is the "Index." Since the Sets are

interlaced, they have a cycle time of 12.8 minutes just as an Epoch. This is not an effective cycle time for a real time data link so a smaller grouping or "Frame" was defined. The Frame is the basic recurring unit of the Link-16 TDMA cycle. A Frame is 12 seconds in duration and contains 1536 Time Slots overall or 510 Time Slots per Set. Since the Time Slots are interleaved, the system can appear as multiple simultaneous communications nets.

Each Time Slot is uniquely identified by Set, Index, and Recurrence Rate Number (RRN). The RRN is the log base 2 of the number of slots assigned to a JTIDS Unit (JU) or group of JUs. This group of slots is defined as a "Time Slot Block." For example, if a JU was assigned a Time Slot Block with all 32,768 Time Slots in an Epoch from Set A, this would be represented by "A-0-15." "A" represents the Time Slot Set, "0" is the starting "Index" indicating that the block starts with the first Time Slot in the Set, and "15" is $\log_2$ 32,768. Since each Time Slot is 7.8125 milliseconds long, the time between the start of successive Time Slots in a Set is 23.4375 milliseconds. TDMA channels assigned enough Time Slots can be used for voice channels. At the other extreme is a Time Slot Block assigned only one Time Slot per Frame (one every 12 seconds). There are 64 Frames per Epoch so one Time Slot per Frame equates to 64

Time Slots per Epoch, and is represented by an RRN equal to six ($\log_2 64$). Therefore A-4-6, B-107-6, or C-433-6 would indicate a JU or group assigned one Time Slot per Frame. The numbers 4, 107, and 433 indicates the sequence within the Frame. Flow control is achieved through Time Slot management. Note a JU can either transmit or receive during any given Time Slot. Voice channels are established by assigning all the voice circuit participants the same Time Slot Block for transmitting and receiving. This is called a contention channel or set up. In this case flow control is achieved by the operators transmit key. [Ref. 24]

Link-16 messages are transmitted in each time slot. Each message contains a header and data. The 35-bit header provides source data and message type. There are four Link-16 message types:

- Fixed format or J-Series
- Variable format
- Free text
- Round-trip timing

Fixed formats are the most commonly used and efficient for exchanging data. They range in size from one to eight 70-bit words (size of words used with Link-16). Most are less than three words. Variable format messages allow users to

send any user-defined message. Free text messages do not have parity checking and may or may not have error correction. Free text messages are used for digitized voice. Round-trip timing (RTT) messages are used to establish and refine net synchronization. A JU transmitting an RTT will actually transmit and receive during the same Time Slot.

Each Link-16 message is transmitted in fixed length 3-word blocks of 225 bits. Each word consists of 75 bits. 70 bits are used for data and five bits are used for parity checks and a spare. The fixed format messages, which are modeled during this project, have three types of words, initial, extension, and continuation. The extension and continuation words are repeated as needed to complete a fixed format message. The initial word contains 57 information bits, an extension word contains 68 information bits, and the continuation word contains 63 information bits. The remaining, of the 70 data bits, are used for labels that describe the message format. A Link-16 message will always be transmitted as a block of three words. If the fixed format message does not fill out the entire three words then no statement words will be used to pad the block.

Fixed format messages are always error encoded with **Reed-Solomon (R-S)** encoding algorithm. This scheme inserts

16 error detection bits for every 15 bits of data or (31,15) encoding and can detect and correct up to an eight-bit error. Error encoding changes the 75-bit Link-16 word to a 155-bit word. After adding the encoded header, a message block containing three-words becomes:

80 bits (header) + 465 word bits (3 X 155) = 545 bits

These bits are encoded with a 32 level symbol (groups five bits per symbol) to create 31 symbols per word or 109 symbols for the header and three words.

The header and data within the Time Slot can be packed in several different ways. Only two will be discussed here, Standard-Double Pulse (STD-DP) and Packed-2 Double Pulse (P2DP). The other packing structures are variations of error control and redundancy that follow the same basic format. Standard packing places the header and three standard Link-16 words into one time slot. Packed-2 Time Slots contain the header and six words. Both use error encoding, a double pulse transmission format (discussed below), a 7.8125 millisecond Time Slot, and can be used with the normal range (300 nautical mile) setting. The primary difference is that P2DP contains six Link-16 words and does not have a jitter period (discussed below).

The Standard Double Pulse Time Slot is composed of five components as shown in Figure 3-1:

- Jitter - Variable (none for P2DP)
- Synchronization - 0.416 milliseconds
- time refinement - 0.104 milliseconds
- message header and data - 2.834 milliseconds

Figure 3-1. Link-16 Standard Double Pulse Time Slot Structure, After Ref. [24].

- propagation guard - At least 1.88 milliseconds

Data is transmitted in the Time Slot as a series of pulse packets. The packet is composed of a 6.4 microsecond pulse and 6.4 microseconds of dead time for a total packet time of 13 microseconds. Each packet represents a symbol of data. In the double pulse modes each symbol packet is sent twice in 26 microseconds to improve jam resistance. There is a single pulse mode available for Packed-2 data packing (not discussed here).

The STD-DP Time Slot begins with a variable dead time called "jitter." This is followed by 16 double pulsed

symbols used for synchronization (0.416 milliseconds) and four double pulsed symbols for time refinement (0.104 milliseconds). P2DP transmits approximately double the data symbols than STD-DP within one Time Slot so the jitter is removed and there is no delay before the synchronization data is transmitted. Next is the message, which consists of a header and the message data. In a Standard packed format this consists of 109 double pulsed symbols (2.834 milliseconds). In P2DP, this consists of 16 header and 186 data symbols for a total of 202 double pulsed symbols (5.252 milliseconds). This is followed up by a dead period to allow for signal propagation to the design range of 300 nautical miles. This requires at approximately 1.88 milliseconds.

To summarize, in STD-DP three Link-16 words with approximately 210 bits of effective data (3 words X 70 bits/word) in a single 7.8125 Time Slot. After overhead, error encoding, parity, and double transmission this message consists of 258 five-bit symbols or 1290 bits. With P2DP, about 420 bits of effective data (6 words X 70 bits/word) are sent per Time Slot. After adding overhead and double pulsing this comes out to 444 five-bit symbols or 2220 bits. The overall data rates are 165.12 kbps and 284.16 kbps respectively.

The Link-16 signal is transmitted over 51 different carrier frequencies in a pseudo-random sequence determined by a seven-bit sequence code (128 combinations) and a hop rate of 33,000 hops per second. This technique is frequency hopping spread spectrum. The 51 Link-16 carrier frequencies are in the Lx-Band, centered three-megahertz apart between 969 - 1206 megahertz. The band between 1030-1090 megahertz is excluded to prevent interference with Identify Friend or Foe (IFF) signals. During a pulse the Link-16 signal uses Cyclic Code Shift Keying (CCSK) to convert a 5-bit code word into a 32-chip sequence called a symbol packet. The 32 possible symbol packets are represented by the phase of a 32-bit Direct-Sequence spreading code, creating the Link-16 Spread Spectrum signal. This makes it possible to recover the original 5-bit sequence in the presence of several chip errors. The carrier is modulated using Continuous Phase Shift Modulation (CPSM) at 5 Mbps using the 32-chip sequence of symbols as the modulation signal. This produces a 5-megahertz chip rate or 200 nanoseconds per chip. There are some additional features of the transmission signal that will not be discussed here. [Ref. 24]

The Link-16 network as modeled for this project is based on the architecture used during a Roving Sands exercise. In the exercise 18 JUs participated over three

Nets using 28 Network Participation Groups (NPGs). For this project, the architecture consists of eight participants, operating on a single Net with Time Slots allocated over three slot groups. The group arrangement was derived from the 1997 Roving Sands Time Slot Allocation sheet. Units operating in different "Sets" of time slots do not interfere with each other, therefore modeling the units operating within the same "Set" can be extrapolated to predict performance of other groups operating within the same set. Reducing the number of participants and slot groups in the model reduces the magnitude of the model without taking away from results.

All participants are assumed within 300 nautical miles and in the line-of-sight (LOS) of each other. As such, no relays were modeled. Link-16 uses a robust spread spectrum signal that resists jamming and employs a powerful error correction code. As such, the assumption is made that mutual interference can be neglected and transmission losses are negligible. These assumptions were made to simplify the Link-16 model.

C. INFORMATION TECHNOLOGY FOR THE 21ST CENTURY (IT-21)

IT-21 is a far contrast from Link-16. IT-21 could be considered a concept of operating commercial off-the-shelf equipment and specifying standards for capacity and interoperability, rather than a specific piece of hardware or legacy communications system. IT-21 takes advantage of asynchronous transfer mode (ATM) technology and high-speed fiber optic networks to provide a robust backbone for networking tactical, logistical, and administrative data.

Bell Labs, as a backbone switching and transportation protocol, developed ATM in the early 1980s. It's a high-speed, multiplexing, and switching technology that transmits information using fixed-length 53-octet (byte) cells in a connection-oriented manner. ATM is the network protocol chosen by International Telecommunications Union (ITU) Telecommunications Standardization Sector (ITU-T) for implementation of Broadband Integrated Services Digital Networks (B-ISDN) [Ref. 25]. The digital techniques used in B-ISDN are capable of handling data, voice, and image transmission concurrently. User-network interfaces (UNI) of 155.52 Mbps and 622.08 Mbps can support high-speed information transfers and various communications modes, such as circuit and packet modes. These capabilities lead to four basic types of service classes:

- Constant bit rate (CBR) emulates a leased line service with fixed network delay
- Variable bit rate (VBR) allows for bursts of data up to a pre-defined peak cell rate
- Available bit rate (ABR) in which capacity is negotiated with the network to fill capacity gaps
- Unspecified bit rate (UBR) allows use of available network capacity, no controls

These tiers of service are designed to maximize the traffic capabilities of the network. The capacity available on VBR and ABR systems will vary. The bandwidth of the UBR class of service is a function of whatever network capacity is left over after all other users have claimed their stake to the bandwidth. CBR is usually the most-expensive class of service and UBR is the least expensive (and most common). As ATM matures, users anticipate that it will provide such advantages as:

- Enabling high-bandwidth applications, including desktop video, digital libraries and real-time image transfer
- Heterogeneous protocols on a single network
- Network scalability and architectural stability

In addition, ATM has been used in local and wide area networks. It can support a variety of high-layer protocols

and is expected to attain network data rates of gigabits per second.

ATM channels are represented by a set of fixed-size cells, identified through the channel indicator in the cell header. The ATM cell has two basic parts: the header (five bytes) and the payload (48 bytes). ATM switching is performed on a cell-by-cell basis using the routing information contained in the cell header. [Ref. 1]

The header information contains the requisite information to facilitate fast multiplexing and routing as well as identifying the type of information contained in the cell payload. Other data in the header performs the following functions:

- Assist in controlling the flow of traffic at the UNI
- Establish Cell Loss Priority (CLP) for the cell
- Facilitate header error control and cell delineation functions

The information in the header makes it possible to transmit ATM cells independently so transmission can be controlled if needed to suit demand and resources. ATM is also connection-oriented. The virtual circuits formed during routing are permanent or semi-permanent, which is better for applications where cell arrival timing is critical such as voice or video applications. [Ref. 26]

The IT-21 configuration aboard USS George Washington (CVN-72) was the original template for the second communication network. An overview of this architecture is shown in Figure 3-2. Due to the complexity of the architecture and difficulty in determining proprietary performance, a simplified architecture was developed and modeled.

The intent of modeling different systems is to provide a means to compare the modeling tools and their utility in a Crisis Action Team environment. It is sufficient then to simplify the network as long as the same template is used for both tools. Instead of modeling the full IT-21 network, the fallback position was to model a generic ATM network (Figure 3-3). This basic ATM network consists of two high-capacity ATM switches (622 Mbps) connected with an optical cable (OC-12). Each switch will support up to six 155 Mbps ATM inputs. To further simplify the network, the six inputs are modeled as one or two ATM edge devises, or LAN Bridges, connecting legacy LANs (Ethernet), and one ATM switch representing input from ATM devices (voice and video will feed through this path). This arrangement will be mirrored at both ends of the network. The legacy LAN inputs (Ethernet) will consist of E-mail servers (e-mail generator) and file transfer (FTP) servers (file generator),

respectively. These generators will represent the Ethernet users sending e-mail and files across the ATM backbone.

The Ethernet hubs will be linked to the ATM edge devices where IP packets will be converted to ATM cells and forwarded to the high capacity ATM switch. Four types of ATM information should be derivable from the higher level (IP) protocol. This ATM information includes source and destination ATM addresses, connection quality of service parameters, connection state, and an ATM virtual circuit identifier which maps to a single application. Only quality of service parameters will be modeled.

Data arriving from the ATM devices obviously does not need to be converted into ATM cells. The model assumes all packets and cells arriving at the ATM edge devices or switches are addressed to the distant end. The edge devices will be linked to the ATM switches via OC-3 cables. Their purpose is to convert the Ethernet packets from an Internet Protocol (IP) to the standard ATM packets then forward the ATM cells to the main switch. The simplified architecture is shown in Figure 3-4.

The main switch will provide access control and Quality of Service functions. The access control and Quality of Service functions are very basic in the model. Data packets will be provided high data integrity but low priority on

packet delay. The voice (ATM) sources will be guaranteed a minimum time delay but not guaranteed packet delivery.

The end-to-end performance of the system is measured from the input of the first ATM device to output of the last ATM device. Therefore, collisions and delays associated with the shared media networks (Ethernet) will be neglected. This simplifies the model and data collection while generating the same throughput as multiple, low data rate sources. The model does not go beyond the functions of the AAL5 layer and the ATM layer. The simulation will generate values for throughput, end to end delays, and utilization. It is not concerned with modeling the details between each layer. The logical models are described in more detail in Chapter V, Models.

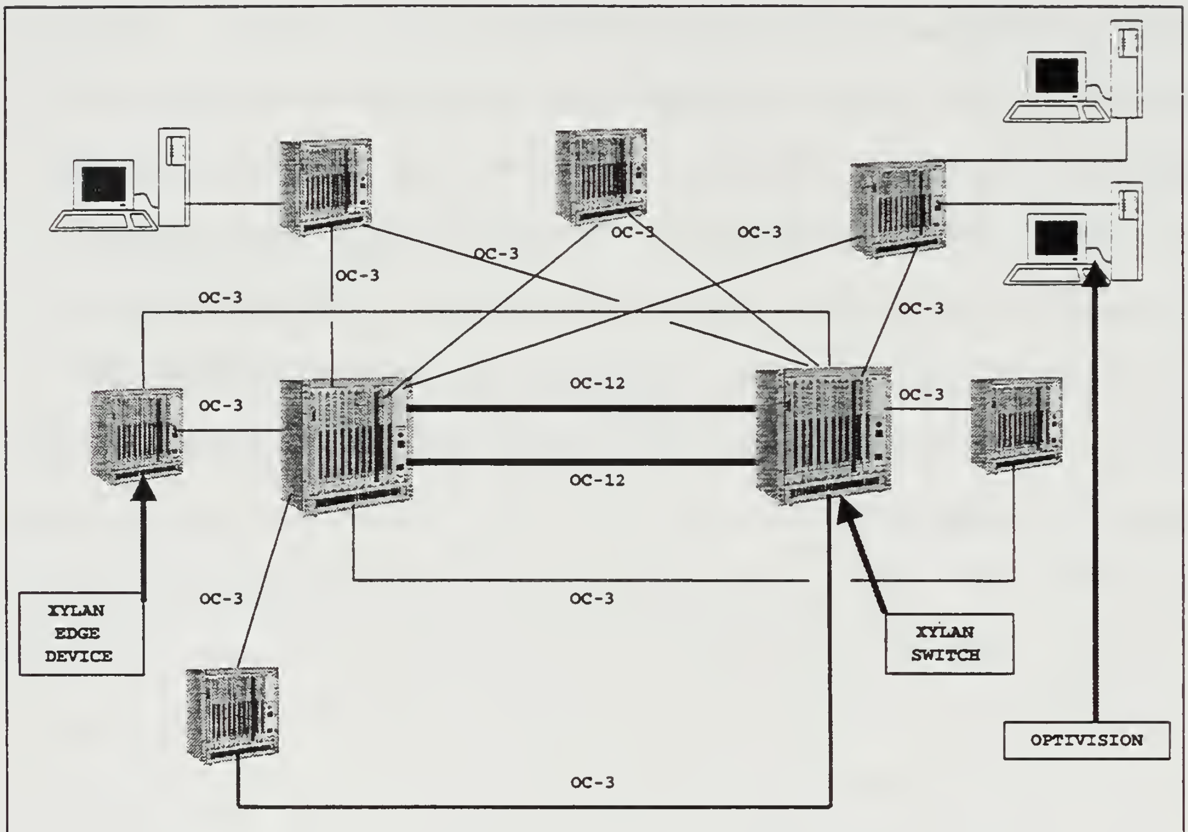

Figure 3-2. IT-21 Configuration for CVN72 From Ref. [27].

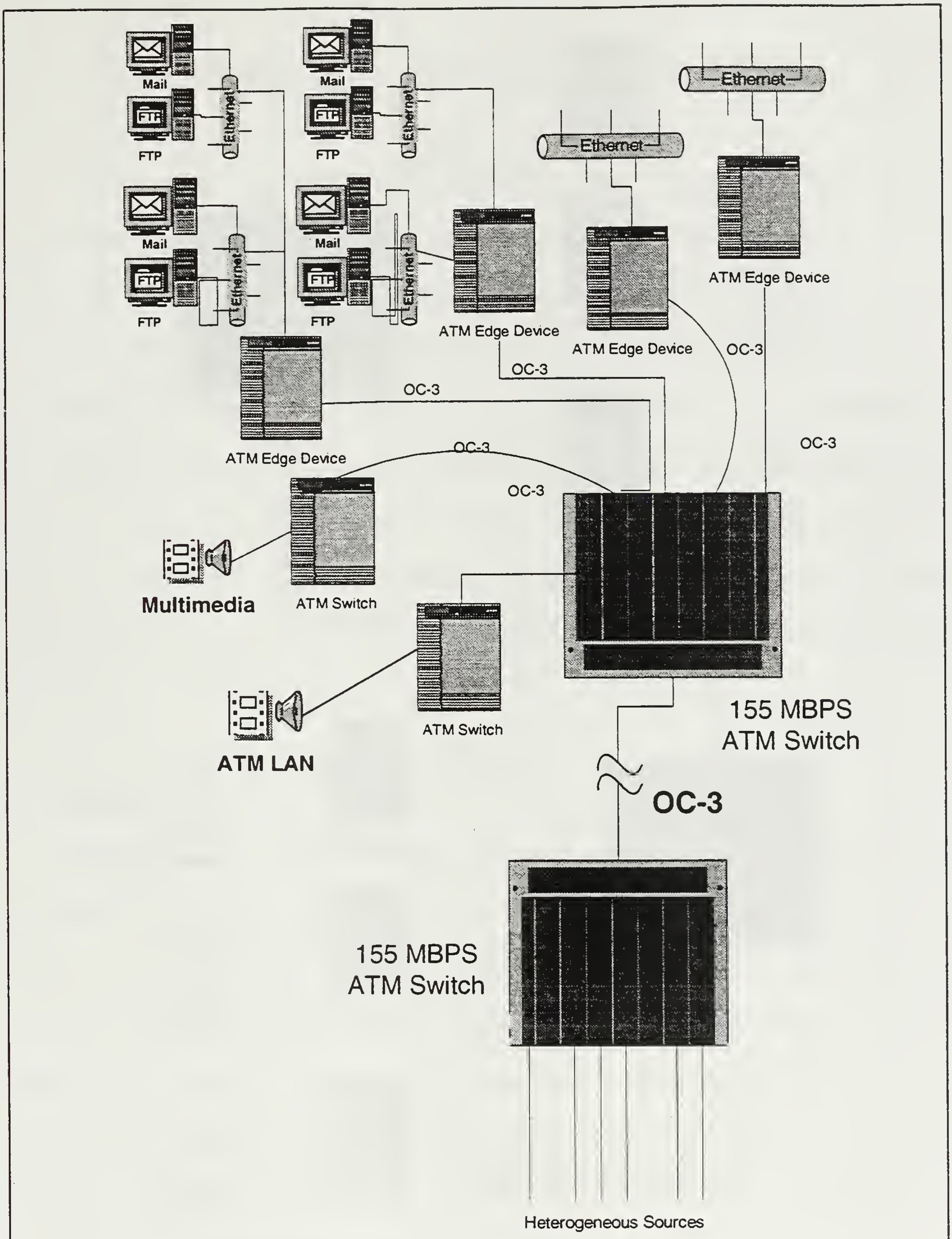

Figure 3-3. ATM Architecture.

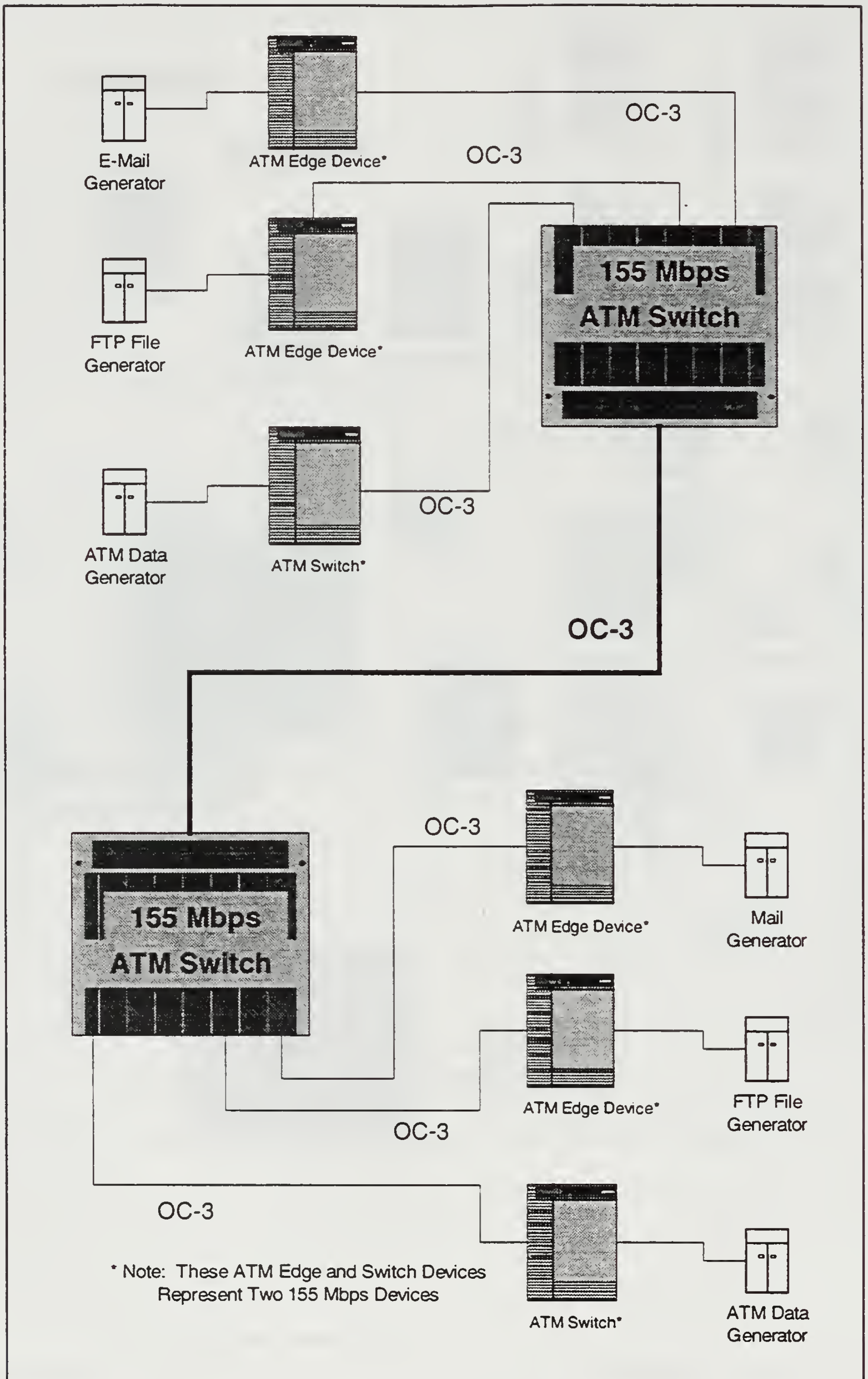

Figure 3-4. Simplified ATM Model Architecture.

IV. MODELING AND SIMULATION TOOLS

Four computer models were developed for this project to simulate the performance of two very different communication architectures. Each communication system selected (see Chapter III, System Architectures) was modeled with two very different automated modeling and simulation tools; **EXTEND**™ by Imagine That!, Incorporated, and **OPNET Modeler Radio** by MIL3. These tools represent the low and high ends of cost and complexity. This chapter expands on the descriptions and capabilities of these two tools that were introduced in Chapter II, Methodology.

A. EXTEND

Extend is an advanced simulation tool designed for decision support. It employs a user friendly graphical user interface (GUI) to develop discrete event or continuous (process) models in a variety of areas. EXTEND can also be used on several different levels. Models can be pre-assembled and distributed for others to populate with data and run. Models can be developed using the many "blocks" or functions shipped with EXTEND. Users can also develop their own blocks or functions by modifying the original blocks or

building new blocks using the built in programming environment called ModL. Larger models can be organized into user selected hierarchical blocks representing subsystems or functions.

EXTEND comes in four configurations. The basic science and engineering configuration was used during this project. It provides complete functionality and 14 EXTEND libraries. Other configurations are essentially upgrades to the basic configuration to provide more predefined blocks or functions. These are the Business Process Engineering (BPR) and the Manufacturing configurations. BPR is useful for analyzing new processes, providing metrics for long range planning, and for modeling organization changes. This package introduces systems analysis techniques to process reengineering efforts. It uses process-flow blocks and has a business-process orientation. The Manufacturing package is tailored for modeling discrete manufacturing, industrial, and commercial operations. Model concepts supported include merging and routing streams of items, batch processes, scheduling, parallel and serial operations, blocking, and closed and open systems. The fourth configuration is a combination of all three packages. [Ref. 7]

1. Requirements

The following configurations represent the minimum requirements to run EXTEND on a desktop computer:

Windows or Windows NT

- DOS 5.0 or later, Window 3.1 or later, and Win32s 1.2 or later or Windows 95 or Windows NT 3.5 or later
- 80386 processor or greater (80486 or Pentium Recommended)
- 4 MB of RAM (8+ MB recommended)
- 10 MB of hard disk space
- Video Graphics Array (VGA) or better graphics capabilities
- Math co-processor recommended

Macintosh or PowerMacintosh

- System 6.0.7 or later
- 68000 processor or greater (quadra or PowerMacintosh recommended)
- 4 MB of RAM (8+ MB recommended for System 7 or large models)
- 8MB of hard disk space

EXTEND has on-line help and Imagine That, Incorporated provides technical support to registered users in several different formats. [Ref. 7]

2. Basic Modeling

Some of EXTEND's more common parts are discussed in this section to provide an overview of building models and running simulations with this tool. The most basic items include the libraries, the blocks, the blocks dialogs, the connectors on each block, and the connections between the blocks.

Libraries are archives for the block definitions (icon, dialog, and code). The blocks are separated into libraries by their function. When a block is put into a model, a reference to the block information in the library is added, not the block itself. If the definition for a block is changed in the library it will update all the models that use that block. The libraries used most often are the discrete event, generic library (for continuous simulations), and the plotter library. Some of the more commonly used blocks from these libraries are discussed below. Other libraries included with the basic package are the animation library, electronic engineering libraries (to simulate analog, digital, signal processing) and sample libraries such as Scripting Tips, Custom Block, and Utilities libraries, that help illustrate EXTEND features. Users can also create their own library to hold user-defined blocks and hierarchical blocks.

Blocks are indeed the foundation of an EXTEND model. They define actions or processes within the model. Each block has six basic parts: dialog, script, icon, animation, connectors, and help text. Dialog allows the user to set a block's behavior and to input or output data. Script is the ModL program or code that makes a block work by selecting the inputs from the connectors and performing the desired operations. An icon that represents its function identifies each block. Animation allows items to be followed during simulation. Connectors are used to input and output data to and from other blocks. The help text describes the block function, dialog boxes, and each of its input and outputs. Blocks can represent sources of information or modify items. Some are a combination of blocks organized to form a higher level hierarchical block. Each block represents a portion of a model, which is assembled like a block diagram. Some of the more commonly used blocks, available in the basic EXTEND configuration, are discussed below.

During a simulation, discrete event blocks pass items or objects between them, performing some type of operation on the item or its attributes. The following discrete event blocks are describe briefly: Generator, Program, Queue, Delay Activity, Timer, Set Attribute, and Make Your Own. Generators provide items at specified intervals (parts,

network messages, etc.). Program blocks, similar to Generators, are used to schedule many items such as a sequence of events. Queues are holding areas for items waiting further processing such as buffers. They also track the time an item spends in the queue and the length of the queue. There are several types of queues; first-in-first-out (FIFO) and last-in-first-out (LIFO) are two examples. The user can set queue attributes such as maximum queue length. Delay Activity blocks are used to hold an item for a specified amount of time such as propagation, processing delays, or net cycle time. Timers are probes within a model and are used to measure the time it takes an items to pass between two points. This is useful to measure end-to-end delays across buffers (queues), edge devices converting packets (activity delays), and propagation delays. Get Attribute is used to access or remove information (values) attached to an item. Attributes could be used to add source or routing information, message type, size, priority, or other information unique to the object. There are several blocks that allow the user to modify an item's attributes. Make Your Own blocks provide the user with a template to create custom blocks. These block have universal connectors and labels, the user just adds the script. EXTEND blocks

are scripted with ModL program language, which very similar to the C Programming language. [Ref. 7]

Generic blocks are used in continuous models and perform special tasks in discrete event models. These blocks help the user avoid programming special blocks. The following blocks, from the generic library, perform most of the basic functions: basic math, input, output, decisions, accumulators, and data conversion. Input blocks include functions to read in data from text files and input random numbers. Decision blocks provide logical operators to make decisions based on user parameters and item attributes. Accumulators can sum or integrate inputs over the course of the simulation. This could be used to determine total throughput and utilization. Conversion tables allow the user to set up math conversions, such as units of measure, or set up a table to convert values such as converting an Ethernet packet to a number of ATM cells.

Dialog items are used to specify block actions or processes. Dialogs are pre-defined for each block and can be used to enter values before and during a simulation. Opening a particular block accesses dialog items. The dialogs can remain open during a simulation to allow the user to change settings or enter new parameters for a block.

Some blocks report values in their dialog and can be used to display values during a simulation.

Connectors are the points on a block were information enters or exits. Connectors are pre-defined to support the function of the block. As such, blocks can have different numbers of connectors depending on the operation it performs. The type and direction of the information passing through them identify connectors. The two information types are **item** connectors and **value** connectors. Looking at the direction of information flow, connectors receiving items or values are called **input** connectors. Values or items are output from blocks at **exit** connectors. For example, an item leaving a block would pass out through an **item-exit** connector. Since values represent an attribute or number associated with an item, **value** connectors can be connected to many different blocks and each block will receive the value, much like a broadcast. However, "items" represent physical entities or objects as they pass through a model. If an **item-exit** connector is linked to several **item-input** connectors then it is possible for that item to be forwarded to any block ready to receive the item but only one block will receive it. This is analogous to a packet going though a router.

3. Running A Simulation

The simulation functions let the user define how the simulation will run. All simulations must have both run time and the number of runs specified. For discrete event models, only the start and end times need to be entered. The number entered corresponds to the number of time units that the model will run. Since extend works in time units, the user needs to make sure all the processes and parameters are based on the correct time unit. For example, if the run time was set as 24 to represent one day, the basic unit is an hour. If a generator block needs to generate an item every minute in this simulation, then the interval would be set to 1/60 vice one. For continuous simulations the user can select the run time and either the step size or the number of steps. If step size is set then the number of steps is calculated from the total run time. Conversely, if the number of steps is specified, step size is calculated.

Data can be imported and exported from EXTEND using text files. This provision allows data contained in a database or spreadsheet to read into an EXTEND data table. There are several methods to handle text files. One technique is to use the File Input and File Output blocks in the generic library. There are also Import Data and Export

Data commands available from the File menu to allow data to be read from or written to dialogs and plotter data tables. Files can be created from models by using the Reporting and Tracing features. There is also a Sensitivity Analysis function that creates a text file to use for analysis. Finally, users can create their own blocks with input and output functions available in the ModL language.

The EXTEND basic package includes several plotting options. Plots provide a graphical output of selected data and a table of all the points in the plot. There are more than ten different pre-defined plots available in the plotter library. Some plots can be used with both discrete event and continuous simulations such as histogram, scatter plots, and the worm plotter. Some of the plots unique to discrete event simulations are the error bars plotter and the multi-sim plotter. These plotters are designed to use with multiple simulation runs for Monte Carlo or sensitivity analysis. The discrete event plotter tabulates and plots up to four inputs, recording both the value and the time for each. Plots for continuous simulations have similar functions for analyzing multiple runs plus a two unique to continuous simulations. The Fast Fourier Transform (FFT) plotter plots the input and the FFT of the data. The user can specify the number of FFT points. There is also a strip

plotter that behaves like a strip chart to monitor the current conditions of a long simulation. The user can select the number of data points to be displayed on the plot. When you run a simulation, the plotter is displayed on the screen

Animation is another form of output. This can be particularly useful when debugging a model. With the animation set, each item can be followed through the model to see if the model is behaving as expected. The simulation can be setup to pause after each animation change occurs. This can expedite trouble shooting in models with several steps between animation changes.

There are also methods to communicate with external devices such as serial port functions and Windows dynamic-link libraries (DLL). This can be useful for transmitting and receiving data over a modem.

B. OPNET RADIO/MODELER

OPNET Modeler is a vast software package with an extensive set of features designed to support general network modeling and to provide specific support for particular types of network simulation projects. Subsequent

sections of this chapter provide more detailed information on these features, as well as other aspects of OPNET. Here are a few of the more significant capabilities of OPNET Modeler:

- Object orientation
- Hierarchical models
- Graphical specification
- Specialized in communication networks and information systems
- Flexibility to develop detailed custom models
- Automatic generation of simulations
- Application-Specific Statistics
- Integrated post-simulation analysis tools
- Interactive Analysis
- Animation

The first four features are similar to those previously discussed with other modeling tools. OPNET uses windows, dialog boxes, buttons, and scroll bars, and the mouse for input whenever possible. OPNET Modeler stands out particularly due to its capability to develop detailed models relating to networks and communications. A somewhat unique capability is the automatic generation feature. Model specifications are automatically compiled into

executable, discrete-event simulations implemented in the C programming language. Advanced simulation construction and configuration techniques that are employed to minimize compilation requirements.

OPNET provides numerous built-in performance statistics that can be collected during simulations. Users can augment this set with application-specific statistics that are computed by user-defined processes. OPNET also includes tools for graphical presentation and processing of simulation output.

Simulation sequences can be configured to generate animations of the modeled system at various levels of detail to include animation of statistics as they change over time.

OPNET can be used to model a wide range of systems. Here are just a few typical applications that OPNET features specifically support:

- Standards-based LAN and WAN performance modeling
- Inter-network planning
- Research and development in communications architectures and protocols
- Distributed sensor and control networks
- Resource sizing
- Mobile packet radio networks
- Satellite networks

- C3I and Tactical networks

Proto-C is the program language used in OPNET. It allows development of adaptive, application level models, underlying communications protocols, and links. Performance metrics can be customized and recorded. Scripted and stochastic inputs can be combined to drive simulations. Queuing capabilities in OPNET make it possible to model sophisticated queuing and service policies. Library models are provided for many standard resource types.

OPNET Modeler/Radio contains specific support for modeling mobile nodes, complete with predefined or adaptive trajectories, radio link models, and geographical information. The satellite specific support includes automatic placement on specified orbits, the capability to generate orbits, and animation products to visualize the configuration. To support command and control network modeling, OPNET provides diverse link technologies with the capability to adapt protocols and algorithms using Proto-C, scripted or stochastic modeling of threats, and pre-defined radio link models.

1. System Requirements

The OPNET program is the most visible part of the OPNET system. The OPNET program is window-based, using the MIL 3 User Interface (M3UI); a GUI similar to those used in other interactive applications. The OPNET window is managed by the workstation's window system, which determines the window's appearance and whether it can be moved or resized. OPNET is GUI-based, it can only be used from a graphics workstation console or X terminal. OPNET cannot be used from an ASCII terminal. See Figure 4-1 for the window systems supported by the OPNET program.

Workstation Type	**Window System**
DEC	DECwindows (X Window-based)
HP	HP Visual User Environment
Silicon Graphics	IRIX X Window System
Sun	OpenWindows (X Window-based)
Any UNIX	MIT X Window System
Windows NT	Native

Figure 4-1. OPNET System Requirements From Ref. [16].

2. Basic Modeling

A network is comprised of physical sites, referred to as nodes, which may originate and transmit information, receive and process information or both. These nodes communicate via links, which may take the form of electrical wire, fiber optic cable, or radio-microwave links. The behavior of nodes is defined by their process attributes and associated parameters. To develop models in this manner OPNET uses a hierarchical structure that separates editing environments for the design of different functional and logical levels. The Network Editor is at the top level. The subordinate hierarchical levels are the Node Editor, Process Editor, Parameter Editor, and features accessed through C language within the OPNET kernel. In this section the Network, Node, Process, and Parameter models will be briefly described with their associated editors. [Ref. 28]

The Network Editor is used to develop all high-level components of a network. The user has access to multiple types of node platforms from within the editor. Each node in a network model represents a particular communication facility. The internal functions of those communication facilities are defined in the node models. The node models are created in the Node Editor. There are no specified

limits on the number of nodes within a network model. The nodes in a network model may communicate with each other via point-to-point links, bus links, or radio links (OPNET Modeler/Radio). These links are graphically added within the network editor except radio links, which are not represented graphically. Radio links existence depends on position, radio frequency, power levels, and other varying attributes that may cause radio links between any radio transmitter and receiver pair to appear and disappear dynamically during a simulation.

As systems become more complex, it can be useful to group several related nodes within a network as a single aggregated unit. In OPNET this grouping of nodes and their links is called a subnetwork or subnet. The Network Editor has a hierarchical editing system. The highest level subnetwork, called the top subnetwork, contains the entire network model. A typical application is a corporate network connecting several buildings. A subnetwork in the top subnetwork view can represent each building. Nodes and links within the corresponding subnetwork then represent the local area networks within each building.

The user may create node objects and build multiple subnetwork objects inside the top subnetwork or read in a pre-built network model. Once a subnetwork is created, its

contents can be viewed via the subnet view, which is readily accessible to the user. Node, link and other subnetwork objects may be added to the current subnetwork so that there may be more than one subnetwork within the top subnetwork and lower-level subnetworks.

OPNET also has geographic data available in the Network Editor. Subnetworks can be laid out on the selected geographic area and grid properties can be added. In the top subnetwork, the grid units are always degrees. In lower subnets, the units can be degrees, meters, kilometers, feet, or miles. This enhances model visualization, especially when working with WANs or satellite communications. Figure 4-2 below shows a top-level view of a network with several subnetworks (one for each of the three cities). A subnet view is shown on the right.

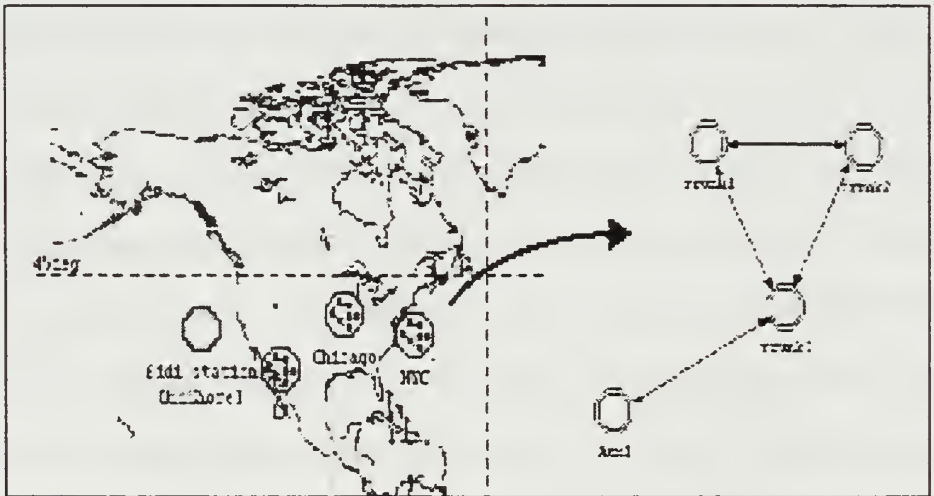

Figure 4-2. Example Subnetwork View From Ref. [29].

In a LAN, each computer and its network interface can be modeled as nodes within a larger network. In a satellite television broadcasting network, for example, nodes might be defined for each satellite, the TV stations that originate the broadcast, earth stations with satellite dishes that uplink and downlink with the satellite, and microwave and cable-based relay stations that boost and retransmit the signal to local receivers. A private branch exchange (PBX) might be considered a node. In general terms, a node is a facility that originates and transmits a signal, receives and processes a signal, or both. Nodes possess at least some of the following internal capabilities in relation to messages in the network:

- Creation
- Transmission
- Reception
- Storage
- Internal routing
- Internal processing

These capabilities represent the functions that a node model needs to provide. The Node Editor provides the resources necessary to model the internal functioning of

nodes through a graphical interface. Within the Node Editor, the user has access to a variety of pre-defined modules. Each kind of module models some internal aspect of node behavior, such as data creation, storage, processing routing, or transmission. A node will usually be made up of several modules. The modules within the node are connected with packet streams or statistic wires. The packet streams carry packets of data, while the statistic wires allow modules to monitor states or status of other modules. This combination of modules, streams and statistic wires allow users to create very detailed models and simulations of nodes. The modules within the node have processes associated with each one of them. These processes can be one of the many pre-defined processes available in OPNET or can be user defined. These guiding processes are called Process Models and are discussed next.

A process can be viewed as a series of logical operations performed on items or data, and a defined set of conditions or rules that guide or direct these operations. In the context of computer and communications systems hardware and software perform these processes. The purpose of the OPNET process models is to model or describe the logical process of the system of interest. Examples include communication protocols, shared resource managers, queuing

disciplines, traffic generators, and more. The Process Editor provides the capability to specify process models. The process models use both graphical and textual components to depict the process. Graphically, state transition diagrams show the logical organization of the process model through icons, to represent logical states, and lines or arcs, to indicate transitions between states. Program statements, based on the C language, perform the actual operations of the process model. Statements can be related to states, transitions, or other blocks within the process model. Script is entered through editing pads provided by the Process Editor. Combining graphics and text have the advantage of providing an overview to understand the process and flow and the power of C language to obtain the flexibility or detail desired within the process. [Ref. 28]

The graphical Parameter Editor provides the recourses to create parameter models. In an abstract sense, a parameter model is a set of data, which characterizes complex properties of objects such as those requiring two or three-dimensional tables. An antenna pattern is an example of a space-varying attribute that requires a three-dimensional table. The Parameter Editor encompasses six parameter models that come with OPNET, which have their own editors. The Probability Density Function (PDF) model

calculates a probability of an action occurring based on a statistical pattern. This can be used to describe packet arrival. The Modulation Functions model determines bit-error-rate (BER) of a digital signal as a function of the effective signal-to-noise ratio. Antenna Patterns model determines the directional properties of antennas. This function can use the antenna patterns and the relative positions of nodes to calculate antenna gain values, which are used to determine received power. The Packet Format model defines the structure or fields within a packet, which are attributes of generator modules found in node models. ICI (Interface Control Information) Format models define the internal structure of ICI's, that are used to control the interrupt-based communications between processes. Link Models specify the attributes for link objects that connect nodes and subnets. Each link object created in the Network Editor becomes an instance of a particular link model. [Ref. 28]

3. Running A Simulation

This section discusses the tools to set up a simulation, run the desired model, record the desired parameters during a simulation, and output and analyze the

results of the simulation. OPNET provides these functions with the Probe Editor, Simulation Tool, and Analysis Tool.

The purpose of developing models and running simulations is to gain insight to a systems performance and behavior. To accomplish this, modelers need to extract the necessary data from a simulation as it runs. Examples of data that could be used to measure network behavior or performance are queue size (buffer), utilization, latency, and throughput. Assuming that the model simulates the desired action or process, the modeler needs to define a set of probes to sense and record the desired parameters. OPNET uses a Probe Editor for this function. The Probe Editor provides the user with eight probe types to collect data. These are:

- Node statistic probes
- Coupled node statistic probes
- Link statistic probes
- Global statistic probes
- Attribute probes
- Automatic animation probes
- Statistic animation probes
- Custom animation probes

The probes can be grouped into three types: statistics, attribute, and simulation probes. Regardless of the type, probes can be thought of as the method of notifying the Simulation Kernel to collect data collection at specific points in the modeled system.

Statistic probes are used for dynamic collection of scalar measurements or quantities such as average queue size, collision rate of packets of a specified link.

Attribute probes are use to record the attributes or values assign to objects or nodes at various levels of the system. Recording attribute values, which are inputs to the model, with the output facilitates comparison of results and analysis. Attribute probes record scalar values.

Animation probes signal the Simulation Kernel to call animation functions. With animation probes, users can animate subnets and see node movement or animate nodes and see packet movement. Custom animation probes activate user defined animation processes.

The Probe Editor display contains three sections: Probe Workspace, Network Subwindow, and Node Subwindow. The user selects the icon-represented probes or creates new probes and places them in the Probe Workspace where they can be edited. The node to be probed is selected from the Network Subwindow, which opens up the Node Subwindow.

Inside the Node Subwindow the user can select a module and assign a probe to it. The user then completes the probe by adding or verifying the remaining probe attributes.

OPNET simulations can be run within the graphical tool or independently using an OPNET simulation utility program. The Simulation Tool allows the user to specify an ordered set of simulation sequences, with different attributes, and execute the simulation sequence. The user defined simulation sequence can be saved as a simulation set object and re-run later. An icon in the Simulation Tool window represents each simulation set. The user must specify any unresolved attributes or may select to use default values before executing the simulation. This is where any attributes that were promoted in the model would have their value entered. Other items specified in a simulation sequence are: the network model, probe file, vector file, scalar file, seed, duration, and update interval. [Ref. 28]

The network model and probe files were discussed previously. These are the files developed by the user in their respective editors. The vector and scalar files are where the simulation results are written. The data put in these files depends on the attributes specified in the probe file, as such both file might not be used. The seed is used for random number generation. The duration and update-

interval specify the simulation run time in seconds and the interval that status reports are displayed during a simulation run, respectively.

Simulations are usually setup to generate data in output files based on the statistic probes determined by the probe model in use. The Analysis Tool is used to pull the data out of the simulation output files (vector and scalar files) and display it using one or more of the plotting methods OPNET provides. Vector files are used to collect data that is dynamic such as a statistic which is changing during the duration of the simulation. Each vector pair contains the value of the statistic and the time it was recorded. Output scalar files collect data this is non-dynamic such as averages, means, and deviations of statistics. The scalars are stored as single values and organized into blocks within output scalar files. The Analysis Tool reads and interprets the data in these blocks to plot the desired metrics. The scalar files can be used to produce plots of Latency verses Load or Latency verses Throughput. Users can plot scalar values as dependent or independent variables. Users can save their plots produced by the Analysis Tool in analysis configuration files to be retrieved later to review plots generated in earlier simulation runs. The plots can also be saved without data

or as templates to be filled with data after subsequent simulation runs. [Ref. 28]

Just as the user can display data in various forms, the Analysis Tool also supports several mechanisms for numerically processing the data, generating new data sets. Examples include calculating cumulative distribution functions, probability density functions, and histograms. Numeric filters constructed from pre-defined filter elements available in the Filter Editor can also operate on the data. A filter model can operate on one or more vectors to form an output consisting of just one vector.

In summary, OPNET provides a comprehensive development environment supporting the modeling of communication networks and distributed systems. Both behavior and performance of modeled systems can be analyzed by performing discrete event simulations. The OPNET Environment incorporates tools for all phases of a simulation study, including model design, simulation, data collection, and data analysis. [Ref. 16]

V. MODELS

This project investigates the utility of using automated modeling and simulation tools in supporting communication planners in crisis action planning situations such as a JTF staff. To this end, four models were developed. Two modeling and simulation tools were used to model a Link-16 network and a computer network based on the IT-21 architecture. EXTEND, by Imagine That!, Incorporated, and OPNET Modeler/Radio, by MIL3, were the tools selected for the project. See Chapter IV, Modeling and Simulation Tools, for a detailed description of these modeling tools. This chapter discusses the four models and model development.

Within dynamic simulation there are two types of modeling methods: continuous and discrete event. In continuous models, time passes linearly and the processes vary directly with time. Examples of continuous-system situations include pollution from a factory and the flow of fluid in a pipe. Discrete-event models deal with events and specific time intervals. Examples of discrete events include computer-performance evaluation and inventory dispatch systems. In discrete-event models, the occurrence

of an event drives the model, whereas in continuous models, the passing of time drives the model. The models presented in this project are discrete-event models. The network models based on the IT-21 computer network are presented first, followed by the two JTIDS (Link-16) models.

A. IT-21 BASED MODEL

As discussed in the project scope, the computer network supporting IT-21 was one of the subjects of the modeling effort. Chapter III, System Architectures, describes the IT-21 computer network and the rationale in reducing the scope of this model. The simplified architecture is based on an asynchronous transfer mode (ATM) wide area network (WAN), comprised of two sub-nets linked by a single 155 Mbps ATM backbone. See Figure 5-1, Top View of Simplified IT-21 Network. Within each sub-net is a group of heterogeneous, local area networks (LANs) running on top of the ATM backbone, Figure 5-2, JTF LAN Topology. Two LANs are 100 Mbps, Ethernet systems operating with a shared medium in a star topology, Figure 5-3, Ethernet LAN Topology. One Ethernet group is designated as the E-mail group and the other as the file transfer protocol (FTP) group. The third

LAN is an ATM LAN representing ATM to the desktop and video teleconference (VTC) capability, Figure 5-4, ATM LAN Topology. The load they generate, E-mail, FTP, or VTC, identifies the Ethernet LANs and ATM workstations. This simplifies data collection when comparing how each tool models the different types of load.

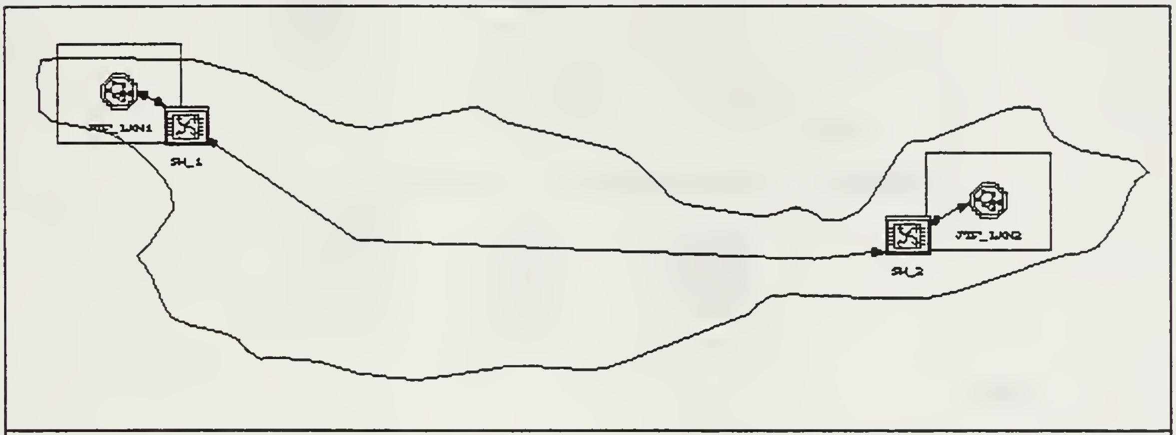

Figure 5-1. Top View of Simplified IT-21 Network.

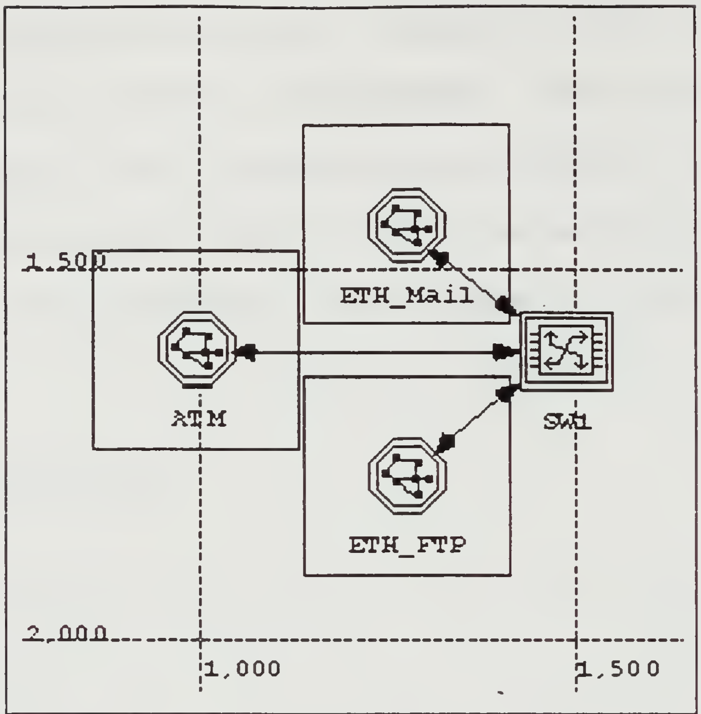

Figure 5-2. JTF LAN Topology.

1. OPNET

OPNET Modeler/Radio is a powerful network-modeling tool. It contains multiple pre-built node and process models representing Ethernet and ATM network components. In the description below, the pre-built models are identified as "OPNET" models or modules. These OPNET models and modules provide the building blocks for this model. Some blocks have several variations and attributes that describe the behavior of the block. Understanding the functions and

attributes of all the node and process modules is critical to building the model.

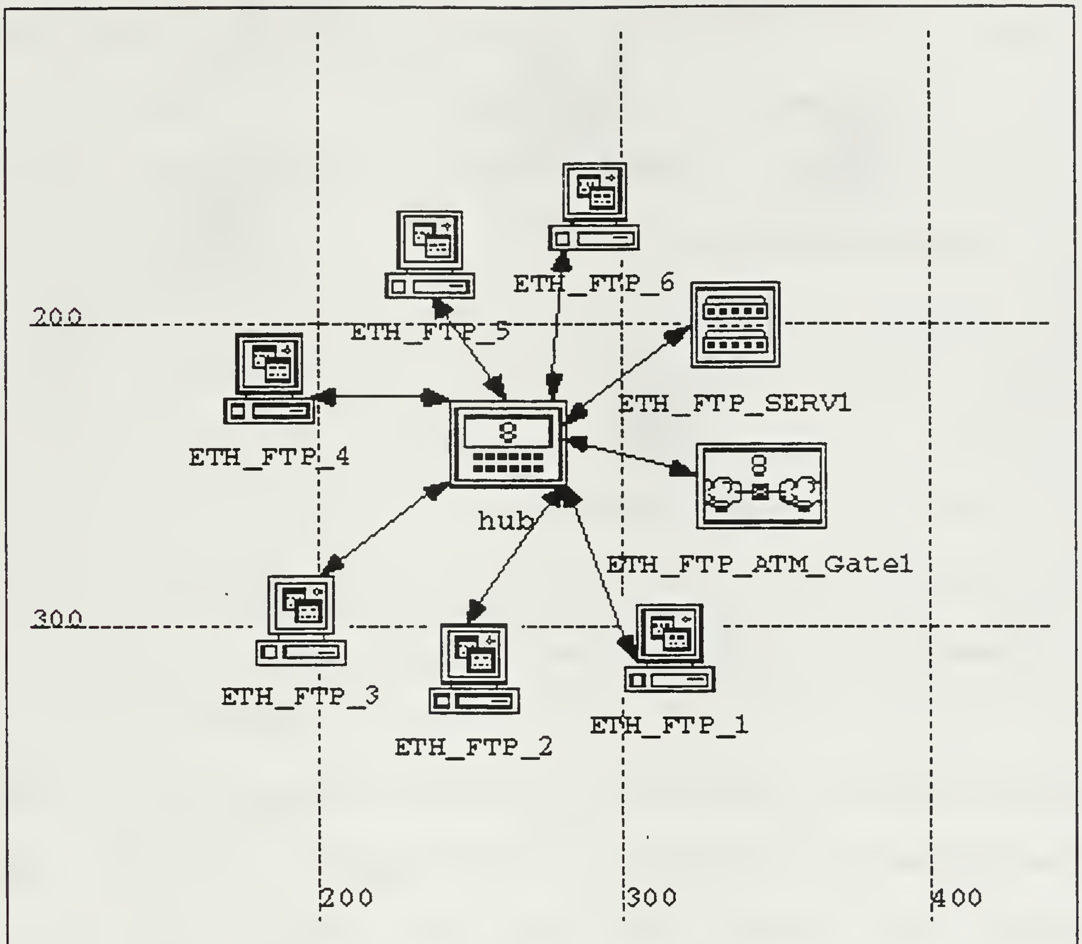

Figure 5-3. Ethernet LAN Topology.

The link between the two sub-nets is modeled with the OPNET 155 Mbps duplex ATM link model. The link connects two 155 Mbps ATM switches at the access point to each LAN. The switches are modeled with the OPNET ATM cross connect node model (Figure 5-5, ATM WAN Gateway Switch). The ATM

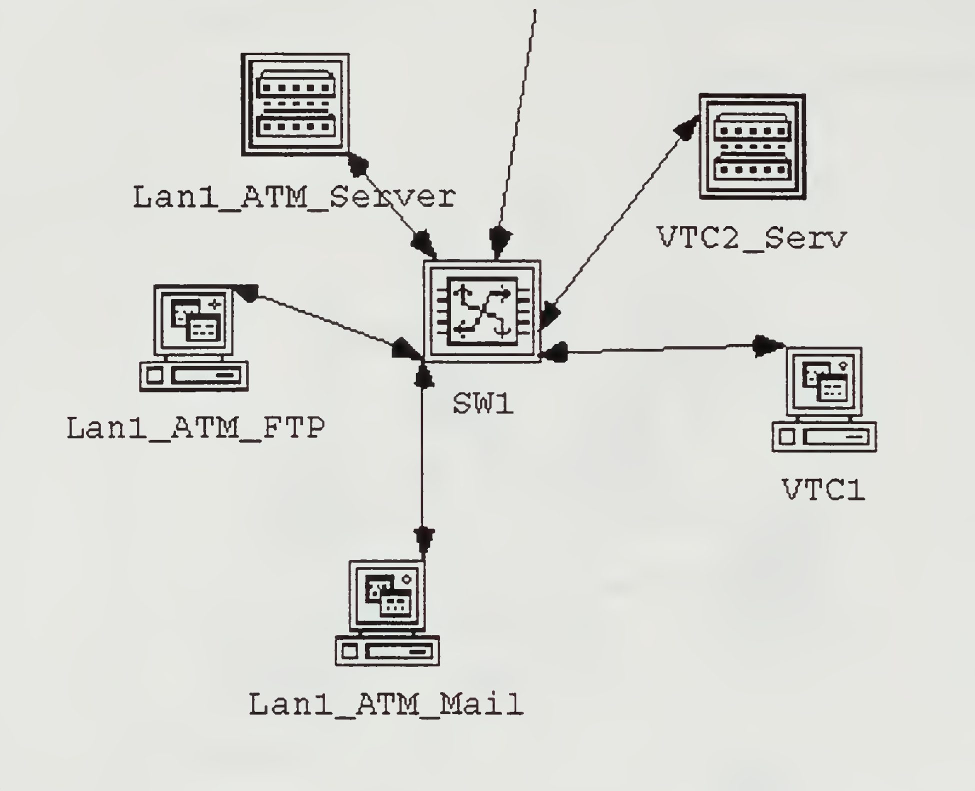

Figure 5-4. ATM LAN Topology.

switches act as routers and traffic concentrators as they perform gateway functions from each LAN to the WAN. The OPNET ATM cross-connect modules provide an option to conduct automatic address resolution and address maintenance, or the user can build their own set of routing tables. An optical transmission interface, such as SONET was not included in the model. If it were, then the payload of the link would be reduced to reflect the additional overhead. For SONET, a payload throughput of 155.52 Mbps is reduced to an effective data rate of 150.336 Mbps [Ref. 1]. This can be modeled

with OPNET by changing the data rate attribute in the ATM data link module to the desired payload rate.

Inside the sub-nets are two identical 100 Mbps Ethernet LANs. Each group has six workstations and a server attached to an eight-port, shared-media hub. The OPNET 100 Mbps Ethernet workstation node (Figure 5-6, Typical Workstation Node) and Ethernet hub models were used to model the workstations and hub, respectively. The workstation node modules contain the attributes that define message generation rate, message size, and their respective distribution functions. The workstation node models also support different client applications, such as E-mail and FTP. These client applications, as modeled, operate over transport control protocol (TCP)-internet protocol (IP), logical link control (LLC), and the medium access control (MAC) protocols. Each Ethernet workstation is connected to the hub with a 100 Mbps data-link, modeled with OPNET's "100BaseT" link model. The two Ethernet LAN 8-port broadcast hubs link to the sub-net's ATM backbone through a LAN emulation client (LEC) or ATM edge device (Figure 5-7, Ethernet-ATM Edge Device). The edge device is modeled with the OPNET ATM-Ethernet gateway node model. The edge device sets up connections to other clients and maps the MAC addresses to ATM addresses. The edge device also segments

the Ethernet packets into smaller, 53 byte ATM cells using ATM adaptation layer protocol type 5 (AAL5). AAL5 provides a connection oriented, variable bit rate service that does not support a timing relationship between the source and destination [Ref. 1]. This means that the ATM cell will contain a 48-byte data segment (payload) and a 5-byte header. The packet segmentation and reassembly rate (SAR) is one of the attributes the user can select. In this model the SAR was set to 8300 packets/second, based on servicing 1500 byte Ethernet packets at 100 Mbps.

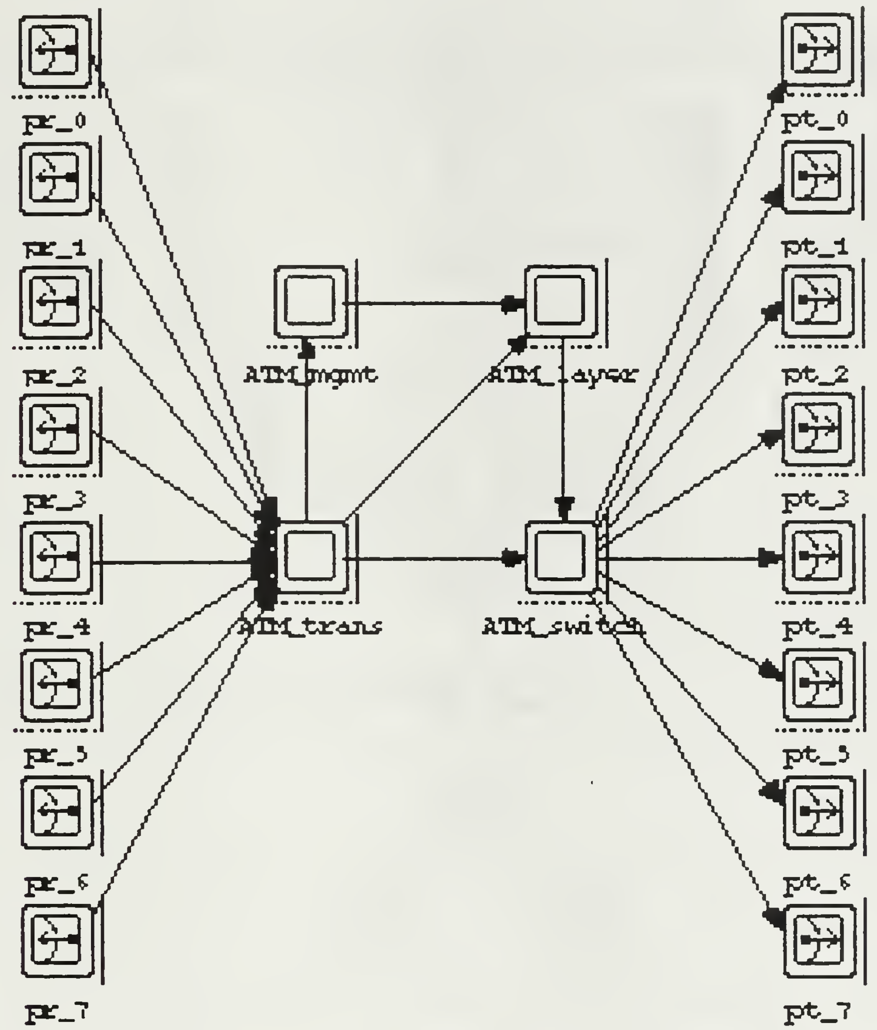

Figure 5-5. ATM WAN Gateway Switch.

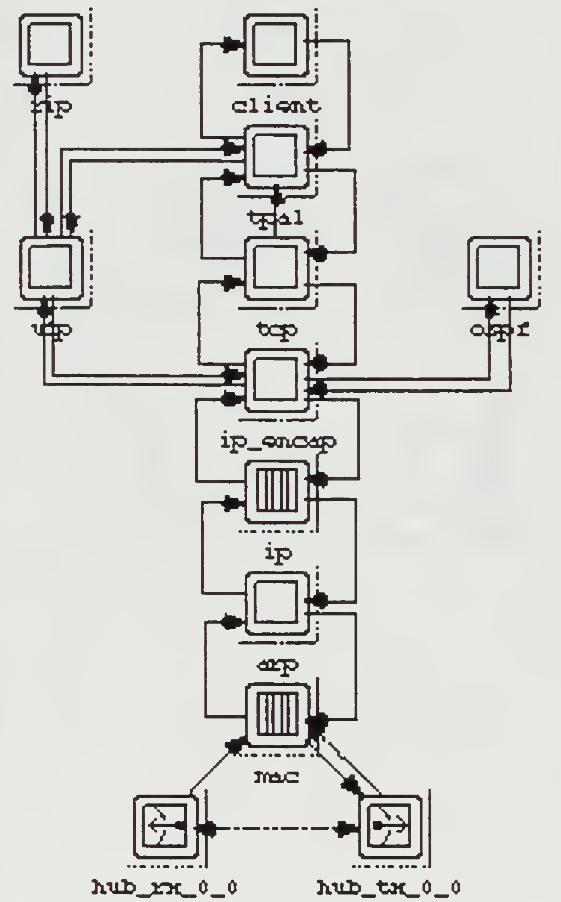

Figure 5-6. Typical Workstation Node.

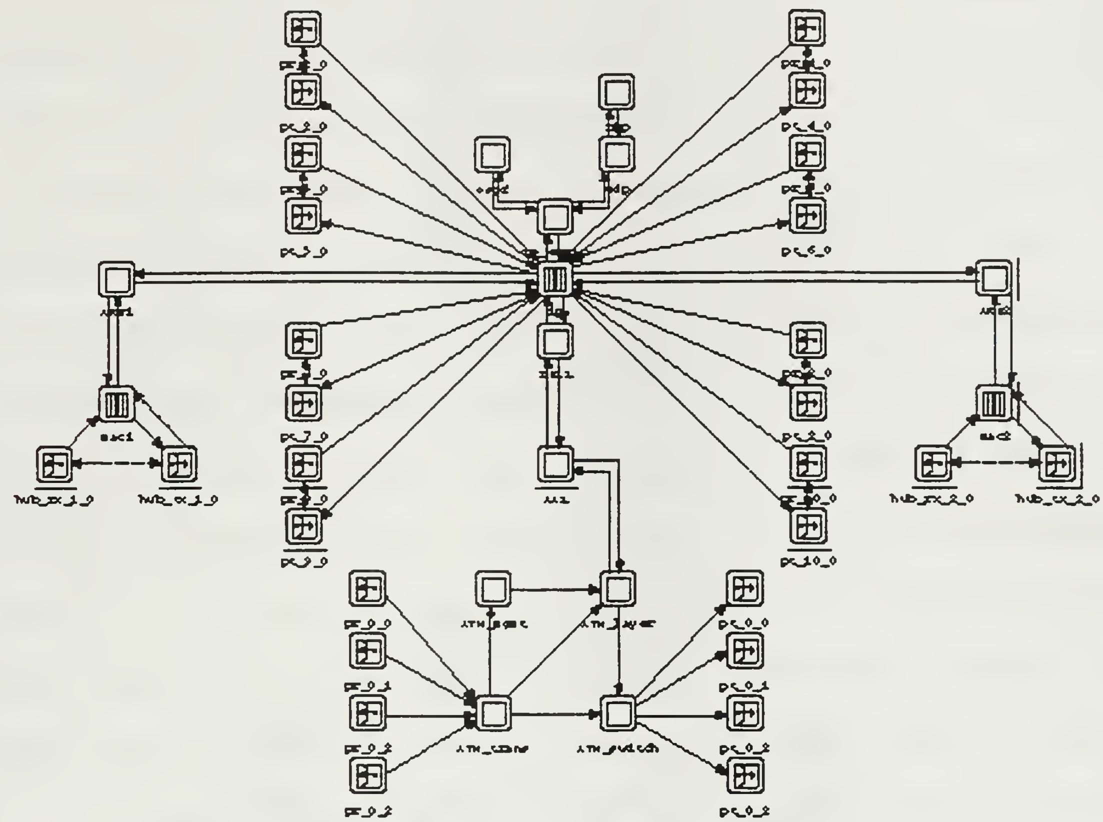

Figure 5-7. Ethernet-ATM Edge Device.

To simplify data collection, one Ethernet LAN was setup with all workstations running the E-mail application and the other Ethernet LAN running the FTP application. The servers on each LAN were set up as E-mail or FTP servers accordingly. In this model, the servers were required to provided control functions such as address resolution. The source type (E-mail or FTP) was used to establish LAN system loading which translates into the workstation message

generation rates and message size. Each workstation was setup to provide above average load since each LAN contained only six workstations. The loading, the same for the OPNET and EXTEND models, is outlined at end of the IT-21 section.

Each sub-net also contains an ATM LAN with three ATM workstations and two servers (Figure 5-4, ATM LAN Topology). Two ATM workstations and one server are modeled with OPNET's TCP/UDP-IP ATM workstation and server node models, respectively (Figure 5-8, Typical ATM Workstation Node). These ATM nodes run client server applications over TCP/UDP. This means these stations will also have selectable SAR values. These nodes represent the ATM E-mail and FTP loads. The server node was set up as the E-mail and FTP server. It was also set to use routing information protocol (RIP) to create routing tables automatically.

The remaining ATM workstation and server were modeled with OPNET's AAL workstation and server node models, respectively. These nodes emulate applications operating directly with the AAL level. This client-server combination represents the video teleconference (VTC) load. The VTC server is also setup to perform automatic address resolution using RIP.

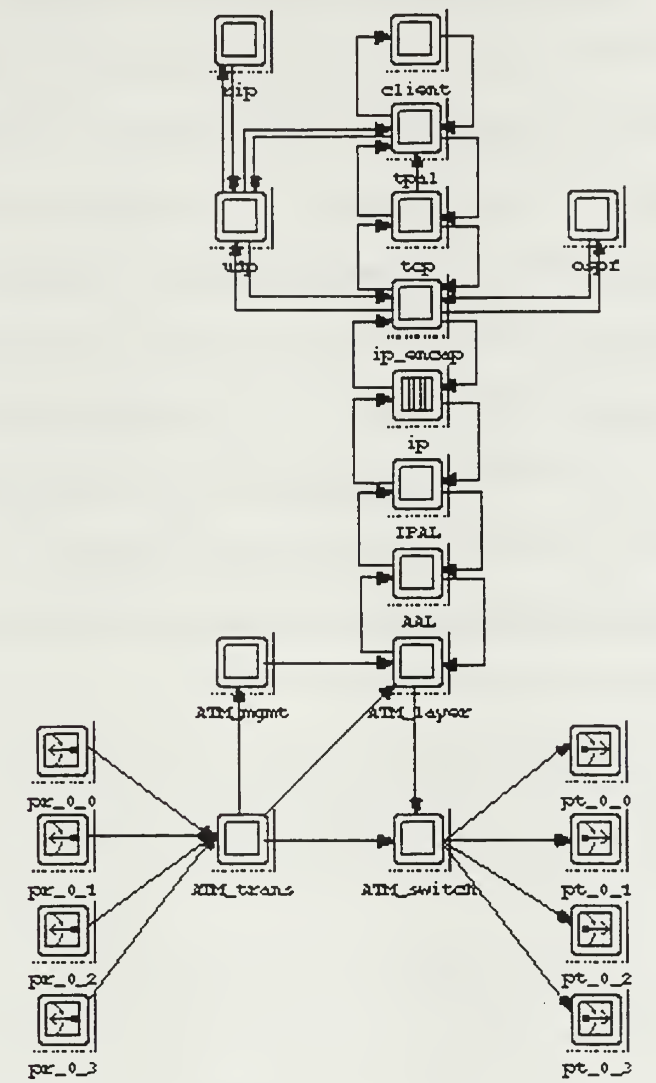

Figure 5-8. Typical ATM Workstation Node.

The nodes in the ATM LAN all link to an eight port, 155 Mbps, ATM switch. This ATM switch, like the others, was set to automatically develop routing tables. The user has the option to manually enter all the routing tables. In the

automatic mode, the switches and servers send out a flood of data units to establish the routing tables. This process was programmed to start at simulation time zero and stop after 5 seconds. This prevents the routing queries from influencing the traffic load measurements. As it turns out, there is an initial flood of data units in the first few seconds of a simulation, then the queries subside and are not a factor in the data measurements. There are several other attributes affecting switch performance. ATM switch priority scheme specifies the priorities within the switch to handle traffic with different Quality of Service (QoS) requirements. The ATM maximum data rate specifies the data rate of a connection. ATM switch fabric delay specifies the delay through the ATM switch fabric. The Usage Parameter Control (UPC) function monitors the connection to determine whether the traffic conforms to the traffic contract. This prevents an overload on one connection from adversely affecting the QoS on another connection. The ATM switch attributes affect system performance. Their settings are summarized in Table 5-1, ATM Switch Settings. Note, ATM switch priority schemes are set to "A" to support VTC data.

Table 5-1. ATM Switch Settings.

ATM Switch Attribute	Setting
Virtual Path (VP) Selection Delay	10E-10
RIP Start Time	0
ATM SAR Rate (packets/sec)	10000
ATM Max Data Rate	155Mbps
ATM UPC Function	Off
ATM Fabric Delay (seconds)	0
ATM Switch Priority Scheme	A

The ATM LAN and the Ethernet-ATM edge device link to the WAN via the ATM cross connect switch that performs the WAN gateway functions.

All workstation nodes in the Ethernet and ATM LANs have attributes to describe message delivery rate and message size (load) during a simulation. The distribution functions for each are called in the workstation process model, which makes it necessary to alter the process model code to change the distributions. Fortunately, the default distributions were desired. Message size is normally distributed. Message arrival rate is modeled as a Poisson arrival rate, which is modeled with an exponentially distributed arrival

interval. The user selects the mean value for arrival rate (messages/hr) and message size (bytes). The workstation set up for the VTC load uses conference interval (conferences/day), frame rate (frames/sec), and frame size (bytes/frame) to describe the VTC load. These attributes are also user selectable.

A probe file was built to collect data during simulation runs. The data is compared to the EXTEND model results in Chapter VI, Analysis. The OPNET probes are listed in Table 5-2, IT-21 OPNET Model Probes.

Table 5-2. IT-21 OPNET Model Probes.

Combined Ethernet LAN Throughput (bps)	**Ethernet Packet End-to-End Delay (ETE) (sec)**
ATM LAN Throughput (bps)	**ATM Cell End-to-End Delay (sec)**
WAN Cross Connect Throughput (bps)	**Ethernet E-mail LAN Hub Collisions**
VTC Throughput (bytes)	**Ethernet-FTP LAN Hub Collisions**

The complexity of OPNET with its myriad of process models, node models, links and other tools can be overwhelming at first. The node, link, and process models selected for this model represent just one way to model this

system. More nodes could have been added for a more realistic model. The goal here is to build two models of the same architecture so the results can be compared. That required knowing more about the settings and attributes in the OPNET models so that a comparison of the results would be meaningful.

2. EXTEND

The EXTEND model development was a sharp contrast to the OPNET model. First the system architecture had to be fully understood. Then, in order to compare the two models, they had to model the same system, using the same attributes or the results could be skewed. To accomplish that task, the OPNET model could not be built in cookbook fashion, instead, it needed to be thoroughly understood. Unfortunately, the models needed to be developed in parallel which resulted in slight variations of what was modeled or what measure of performance was actually measured. Known discrepancies will be addressed in Chapter VI, Analysis.

As mentioned earlier, the EXTEND model is a discrete event model. The approach to modeling the simplified IT-21 network was to simplify the system into smaller, more manageable sections, using traffic flow to identify logical divisions. The IT-21 or ATM network consisted of full

duplex links and switches up to the Ethernet-ATM edge devices. The first division was to separate the problem into two, unidirectional data-flow systems. In this model, the flow is from Sub-Net1 to Sub-Net2 via the WAN ATM link. This makes it possible to model the flow across the WAN but not the data passed between LANs in a sub-net. The model assumes that all traffic generated by a workstation is destined for a workstation in the opposite sub-net (LAN). The flow between LANs within the sub-net could be modeled as a separate architecture in much the same way as this model but that is beyond the scope of this project. In this model the two sub-nets are identical so the model of traffic flow in the opposite direction becomes the mirror image except for traffic load. Another assumption is necessary because the Ethernet LANs use a shared medium and are not full duplex. Here, the model is concerned about the E-mail and FTP loads to the ATM LAN from the Ethernet and not the load within the star. Additionally, with an Ethernet load of about 1 Mbps, the shared medium should appear as a duplex link. To support this, probes were installed in the OPNET model to measure hub collisions. The highest collision rates during the simulations were less than one every 10 seconds.

The directional system was further divided into message sources, protocol routers, switches, and sinks (receivers) (Figure 5-9, Level-1 View of Simplified IT-21 Model in EXTEND). These correlate to a workstation sending data, edge devices, ATM switches, and workstations receiving data.

The Ethernet message generators (Figure 5-10, EXTEND Ethernet Message Generator) were set up to take the user's inputs to produce items (messages) with an exponential arrival interval (Poisson arrival rate). Message size is generated using a normally distributed random number generator. Each item would then be tagged with attributes such as protocol, message size (bytes), and quality of service. Before leaving an Ethernet workstation, the number of packets to carry the message is calculated. This uses the maximum transmission unit (MTU) attribute selected by the user. The message size and system data rate is used to determine a delay time for the message. The priority of the message is set by the QoS attribute, the message is held for the calculated time, it exits the message generator for the Ethernet hub. The Ethernet workstations are attached to an eight node hub which consists of a first-in-first-out (FIFO) queue and "funnels" to produce a single output using the objects from EXTEND's discrete event library (Figure 5-11, EXTEND Ethernet Hub). The queue size is set to buffer 32 Mb

of data. The hub outputs are linked to the Ethernet-ATM edge device. There are no delays associated with the hub.

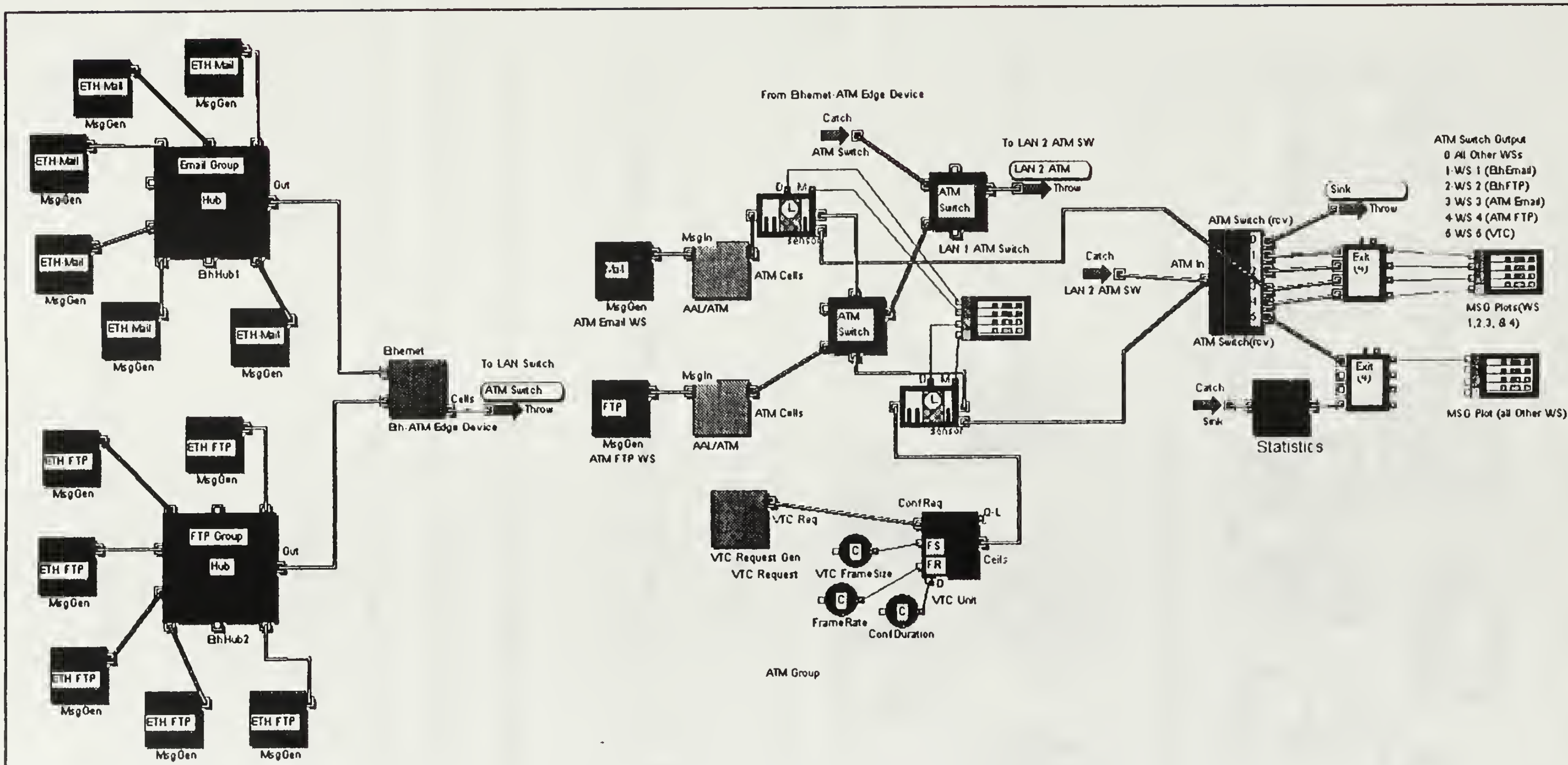

Figure 5-9. Level-1 View of Simplified IT-21 Model in Extend.

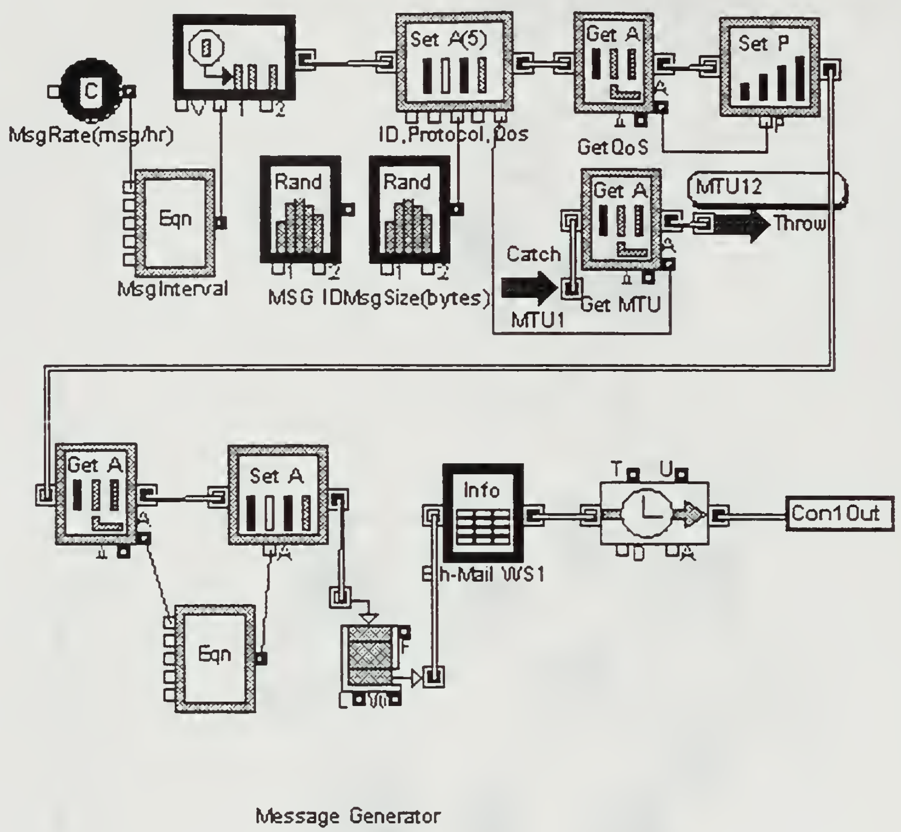

Figure 5-10. EXTEND Ethernet Message Generator.

The Ethernet-ATM edge device (Figure 5-12, EXTEND Ethernet-ATM Edge Device) converts the incoming message items to multiple fixed size, 53 byte ATM cells by using the number of Ethernet packets calculated in the workstation block.

ATM Cells = integer ((# packets * MTU /48) +1)

This conversion assumes type 5, ATM adaptation layer protocol (AAL5) with a 5-byte header and 48 bytes of data. The value is rounded up to the next higher integer to account for ATM cells being a fixed size. The value of the item is then set to the number of cells and sent to a queue. Inside the queue, the item is copied into a number of clones equal to the "value" tag attached to the incoming cell. Each clone retains the attributes and priorities of the original item. Note, attributes, priorities, and values are unique features of an item. Each cell is then delayed for a time based on the edge device segmentation and reassembly rate (SAR) and MTU; parameters set by the user. The data transmission time is considered part of the SAR.

Cell Conversion Delay (seconds) = 48/(MTU * SAR)

Each item exiting the edge device represents a 53 byte, ATM cell with attributes identifying the cell's source, priority, and message originator.

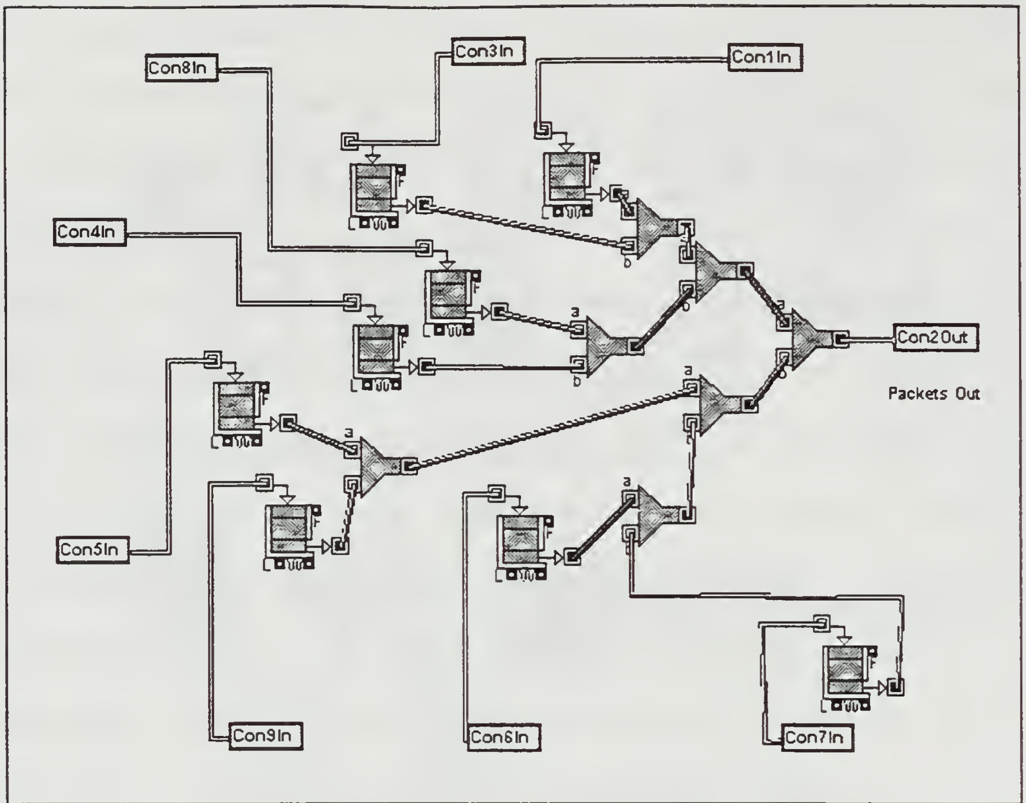

Figure 5-11. EXTEND Ethernet Hub.

The ATM workstation functions are performed in two blocks; the message generator, and the AAL/ATM block. The message generator block (Figure 5-13, EXTEND ATM Message Generator) is similar to the Ethernet workstation except the ATM blocks operate at 155 Mbps data rate and there are no time delays before entering the AAL/ATM block. The AAL/ATM block (Figure 5-14, EXTEND AAL/ATM Block Diagram) represents the AAL/ATM layer of the workstation where the message is segmented into ATM cells and transmitted. Here, each cells is delayed for a period consistent with the system data

rate. The items leaving the AAL/ATM block represent ATM cells with attributes identifying the source, priority (QoS), and original message size.

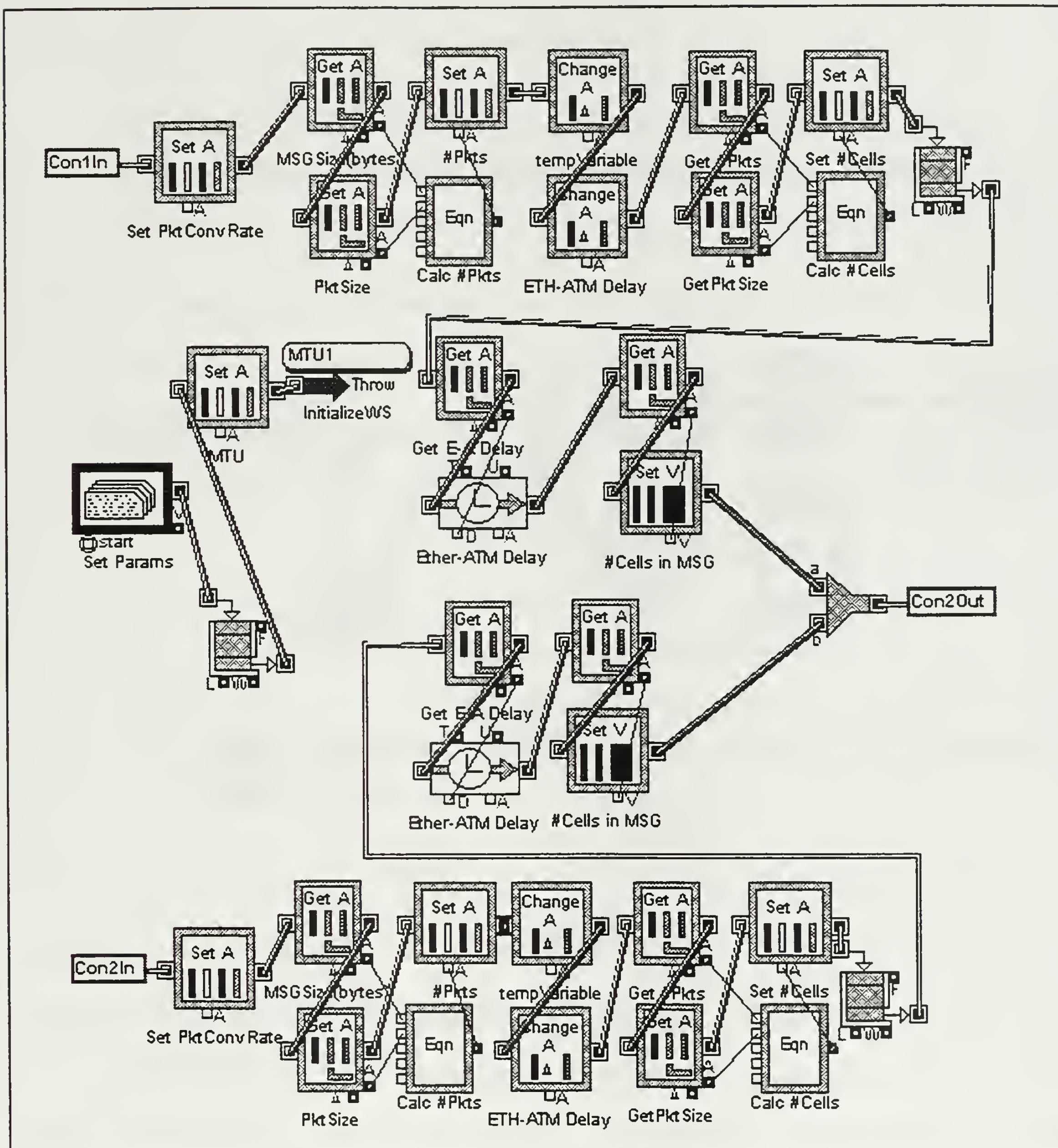

Figure 5-12. EXTEND Ethernet-ATM Edge Device.

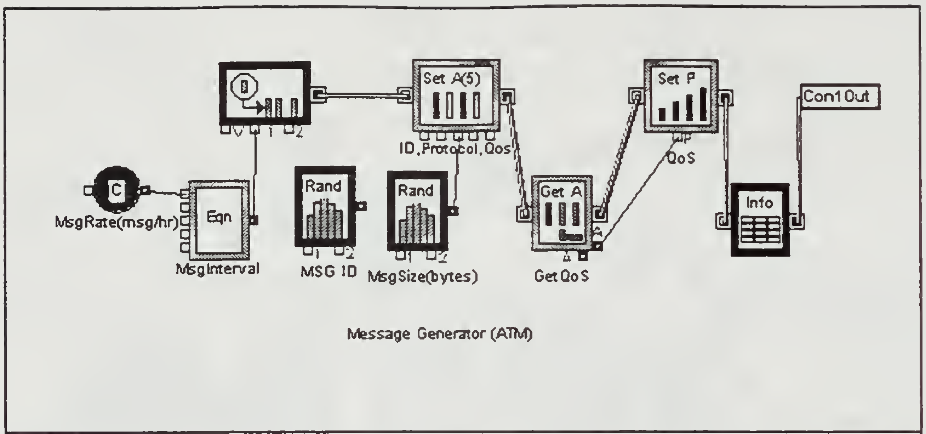

Figure 5-13. EXTEND ATM Message Generator.

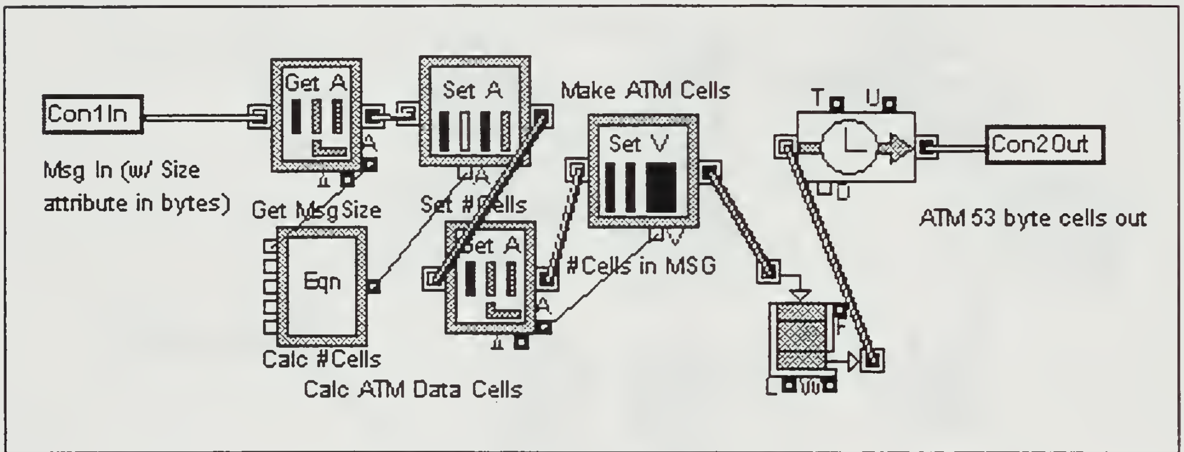

Figure 5-14. EXTEND AAL/ATM Block Diagram.

The third data source is the video teleconference (VTC) group. The VTC request-generator block (Figure 5-15, EXTEND VTC Request Generator Block Diagram) take the user entered data, conference rate (conferences/day), and converts it into a conference interval. This interval establishes the mean for a Poisson distributed conference generation rate.

Conference requests trigger a VTC, defined by duration of the conference, frame rate (frames/second) and frame size (bytes/frame). The VTC, generated by the VTC Unit (Figure 5-16, EXTEND VTC Unit), is represented by a series of ATM cells (items), transmitted at a fixed rate determined by frame size and frame rate.

ATM Cell Rate = Frame Rate * Frame Size / 48

The priority of each cell is set according to the VTC QoS selected by the user. In this model a QoS of 0 is the highest priority, equating to ATM service class "A." All the simulations executed with this model had the VTC QoS set to service class "A" to emulate a constant bit rate, connection oriented transfer.

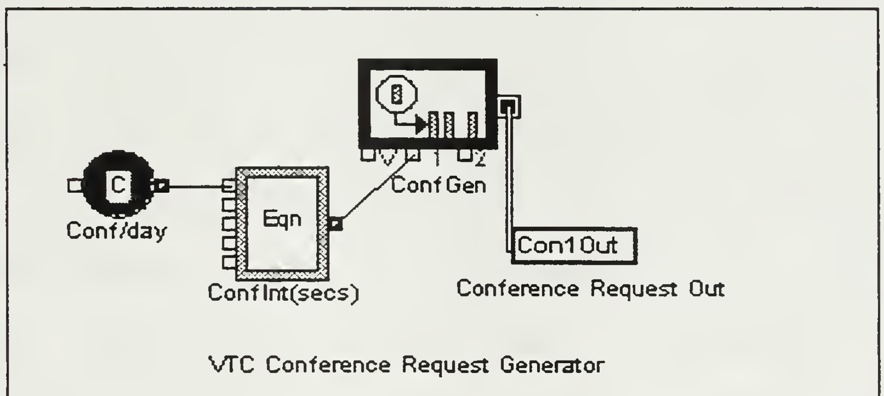

Figure 5-15. EXTEND VTC Request Generator Block Diagram.

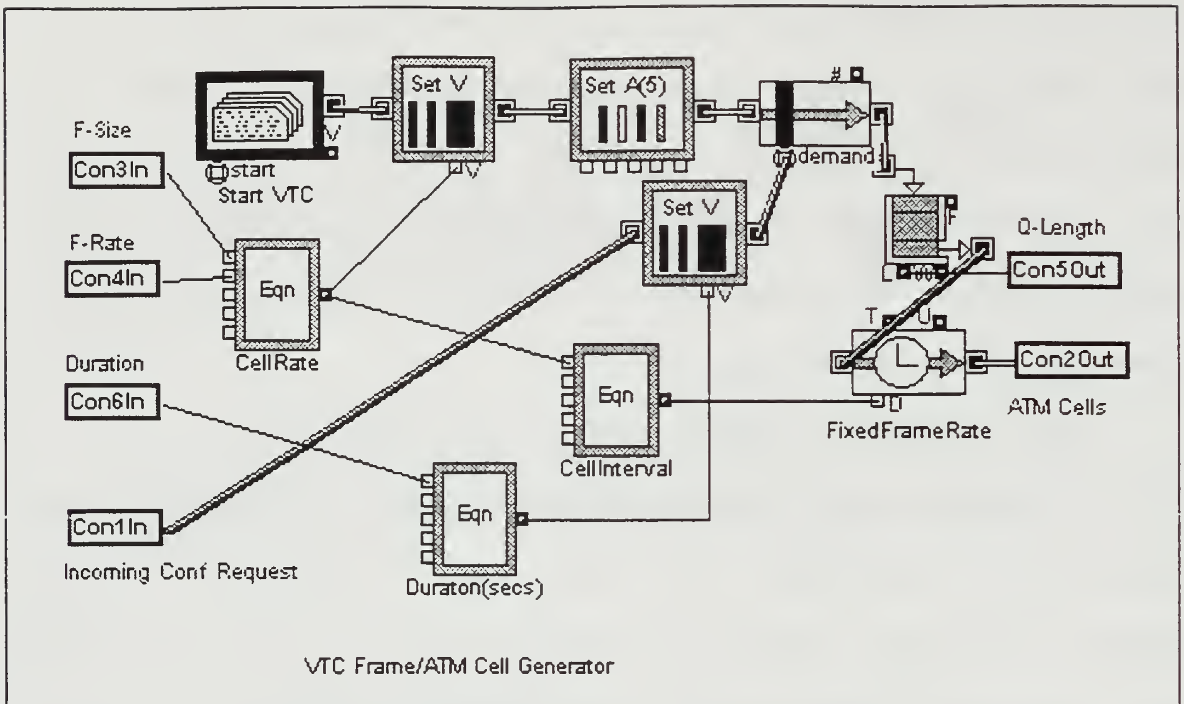

Figure 5-16. EXTEND VTC Unit.

The ATM LAN sources enter an ATM switch (Figure 5-17, EXTEND ATM Switch) that is set to ATM priority class A, which gives priority to the VTC cells if present. This is achieved with a priority based queue which will send the higher priority cells to the front of the queue. The switched ATM cells and the cells from the Ethernet-ATM edge device all forwarded to the ATM WAN switch where they are multiplexed and routed to the receiving ATM switch representing the distant end LAN, Sub-Net 2.

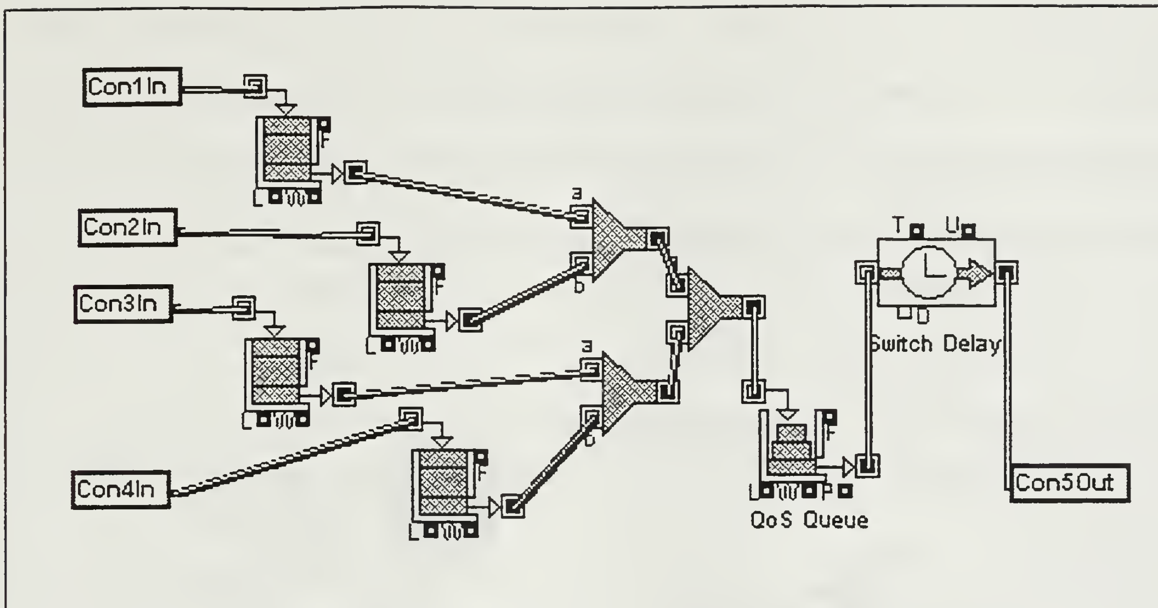

Figure 5-17. EXTEND ATM Switch.

All ATM switches in the model support ATM QoS class A requirements discussed earlier. Each switch also introduces a transmission time delay per cell and virtual path switching delay of 10^{-10} seconds.

Cells arriving at the distant end LAN ATM switch (Figure 5-18, ATM Receive Switch), are switched to one of six distribution points for data collection. Note, the cells are separated by the their source identification attribute to evaluate the message size, throughput and time delays associated with the different messages sources.

The parameters and values associated with both IT-21 models are listed below. If OPNET parameters are not addressed in this section then assume the default value for

the attribute was used. Message and VTC parameters are listed in Table 5-3, IT-21 Simulation Parameters. These values represent the load generated by each workstation. System load will be discussed in Chapter VI, Analysis.

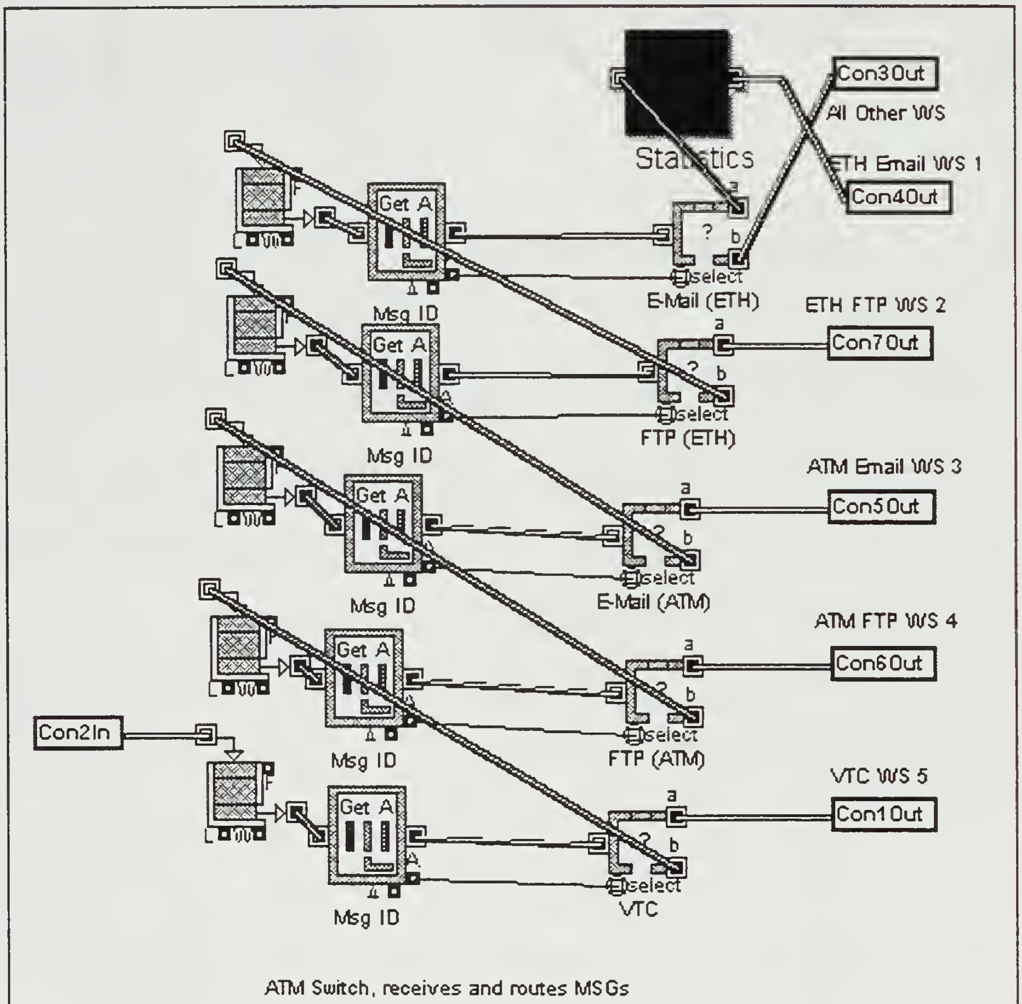

Figure 5-18. ATM Receive Switch.

Table 5-3. IT-21 Simulation Parameters.

Attribute	Value
SAR	**8300 packets/sec**
MTU	**1500 bytes**
E-mail Generation Rate (all sources)	**7200 messages/hour**
E-mail Message Size	**2000 bytes**
Message Size Deviation	**200 bytes**
File Transfer Rate (all)	**3600 messages/hour**
FTP File Size	**50,000 bytes**
Files Size Deviation	**5000 bytes**
VTC Conference Rate	**1 and 240 conferences/day**
VTC Conference Duration	**4 minutes**
VTC Frame Rate	**30 frames/second**
VTC Frame Size	**100,000 bytes/frame**

B. LINK-16

This section describes the OPNET and EXTEND models of a spread spectrum, time-division multiple-access (TDMA), radio data link called Link-16 or JTIDS. Chapter III, System Architecture, provides a detailed description of Link-16. The models here are based on an eight-unit JTIDS net operating in a non-contention mode. The network line up, referred to as slot group assignments, uses a line up from Exercise Roving Sands as a baseline. The network has been modeled for all JTIDS units (JUs) operating in communications mode-1 and TDMA Range-normal.

1. OPNET

The OPNET JTIDS network model contains eight JTIDS node modules, representing the eight JTIDS units in the net. The nodes are located along the Texas coastline using the cartographic views available with OPNET (Figure 5-19, JTIDS Network Top View). All units are within a 300 nautical miles diameter circle and an altitude of 4000 meters. All units will remain within line-of-sight of each other for the purpose of this model. The location of node icons on the network editor grid determines the unit's location in the

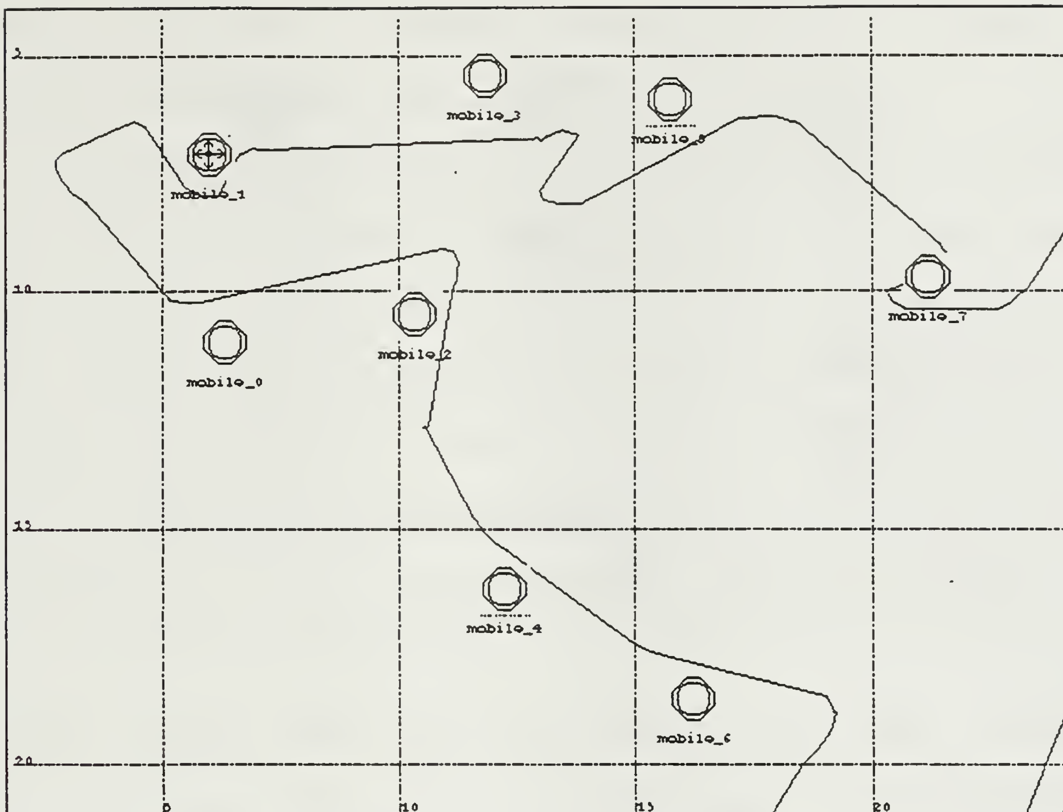

Figure 5-19. JTIDS Network Top View.

simulation. The altitude of each unit is set in the antenna module, discussed later.

Each node model contains five modules, which describe the JTIDS radio equipment and message processor. These modules are antenna, radio transmitter, radio receiver, JTIDS queue module, and data processor module (Figure 5-20, JTIDS Unit). The antenna is modeled with the default isotropic antenna model, with 0 dB receiver gain, and operating at an altitude of 4000 meters. The default

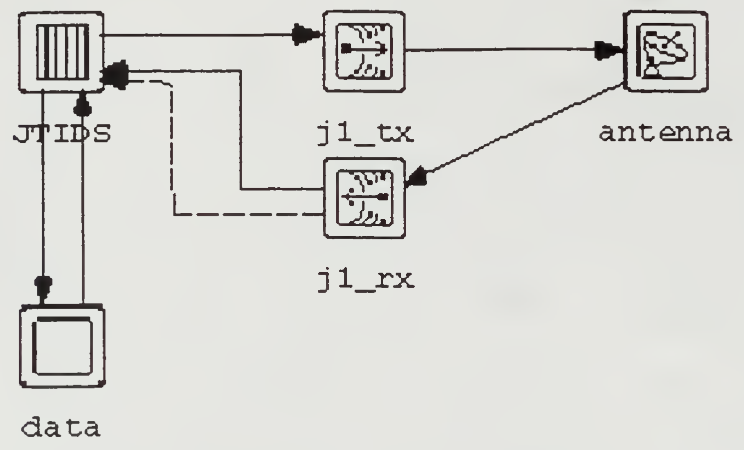

Figure 5-20. JTIDS Unit.

receiver model was used for the model gain, power, background noise, signal to noise ratio (SNR), and bit error rate (BER).

The transmitter module uses the OPNET default channel matching gain, closure, propagation loss, and transmission delivery models. This should result in good radio links with negligible bit error rate.

An assumption is that the transmitter and receiver performance is adequate for the ranges, transmitter power, and the robust error correction associated with the JTIDS signal. Since reception is not an issue in this scenario, the model represents the complex, spread spectrum, modulation scheme, used with JTIDS, with a simple, binary phase shift keying modulation module for the receiver and

transmitter models. The BER associated with reception is assumed negligible.

The JTIDS queue and data process modules are unique to the JTIDS node models. The JTIDS queue process module (Figure 5-21, JTIDS Queue Process Model) is based on a first-in-first-out queue with interrupts to process outgoing packets from the data processor and to forward outgoing "queued" packets to the transmitter at the proper time. This module uses the time slot data, JTIDS set, index, and rate redundancy number (RRN) to control flow to the node data processor and the transmitter.

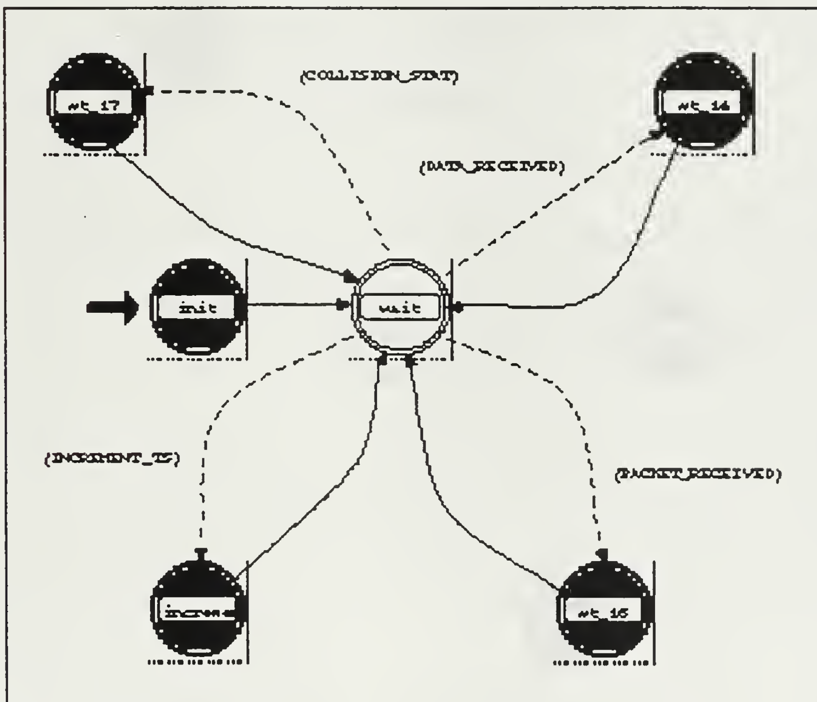

Figure 5-21. JTIDS Queue Process Model.

The data module is a processor module that uses a unique JTIDS process model (Figure 5-22, JTIDS Process Model). The "JTIDS Process" process model monitors packet receipts, maintaining a packet counter, and generates outgoing traffic. Traffic generation is at a rate distributed normally with a mean value of one second and a standard deviation of 0.5 seconds. The outgoing message size is a constant 1000 bits. SPAWAR Systems Center, San Diego, California provided the queue and process models.

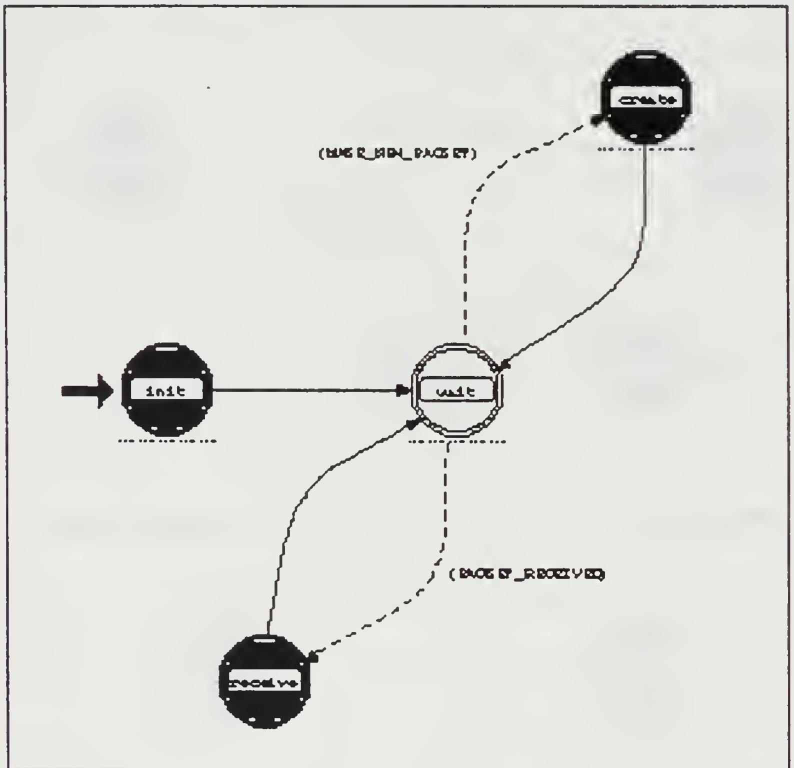

Figure 5-22. JTIDS Process Model.

The node model interface attributes contain the node set, RRN, and start slot or index parameters. The nodes were modeled as JUs operating on the same net, which means they are using the same pseudo random spreading code, generating the same frequency hopping pattern, making it possible to receive each units signals. Within this single net there are three different TDMA schedules or slot groups. Four JUs are in one group and two JUs are in each of the other two groups. In each slot group, the JUs are assigned a specific set of JTIDS time slots for transmitting data. These are called slot group assignment and they are composed of a "Set," index, and RRN as described earlier.

2. EXTEND

The EXTEND model of the JTIDS network is made up of eight objects at the top layer (Figures 5-23, JTIDS Network Model in EXTEND), each representing one of the JTIDS Units (JUs). Each JU module or block (Figure 5-24) contains a transmitter processor, receiver processor, and transceiver block.

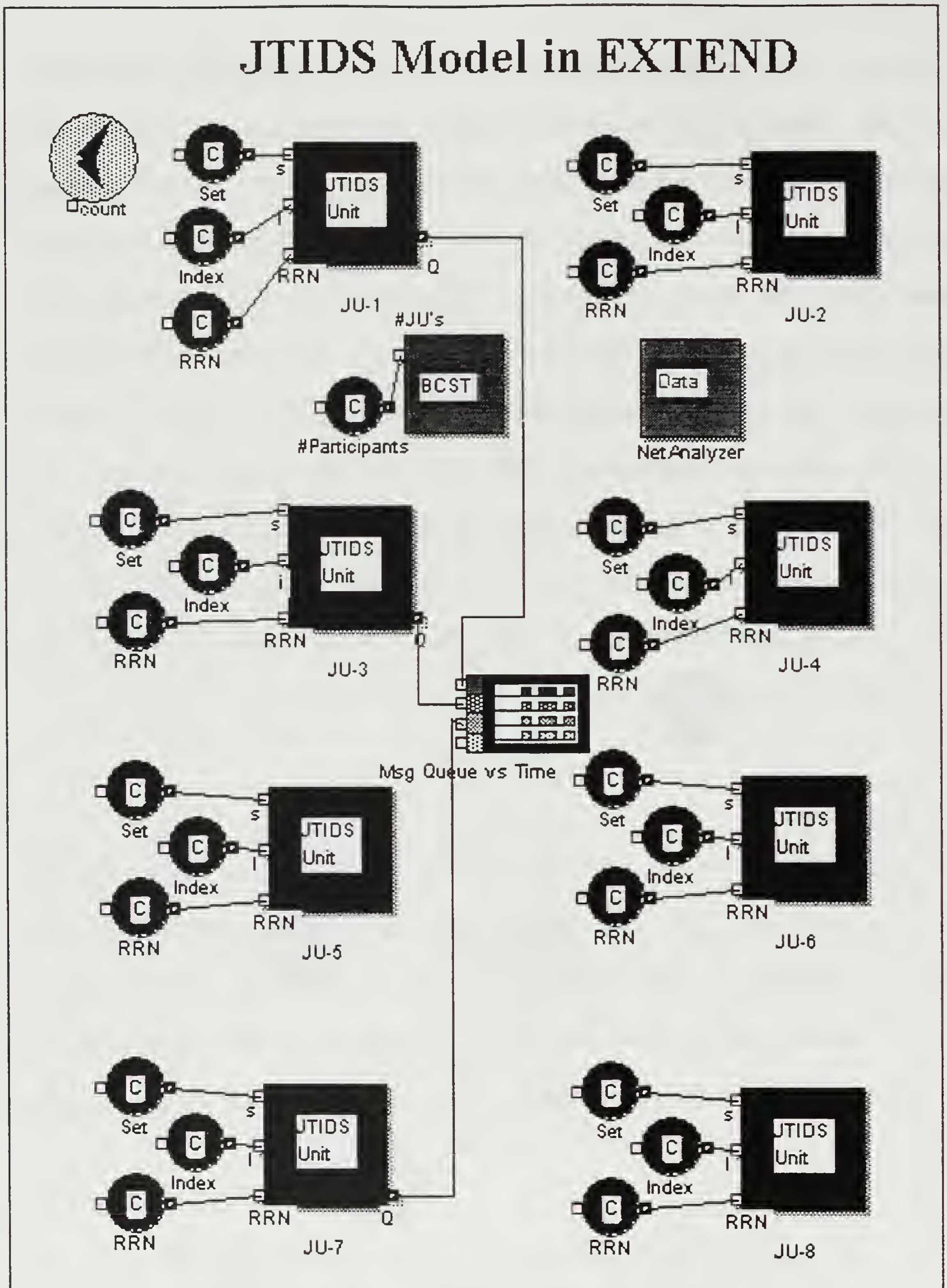

Figure 5-23. JTIDS Network Model in EXTEND.

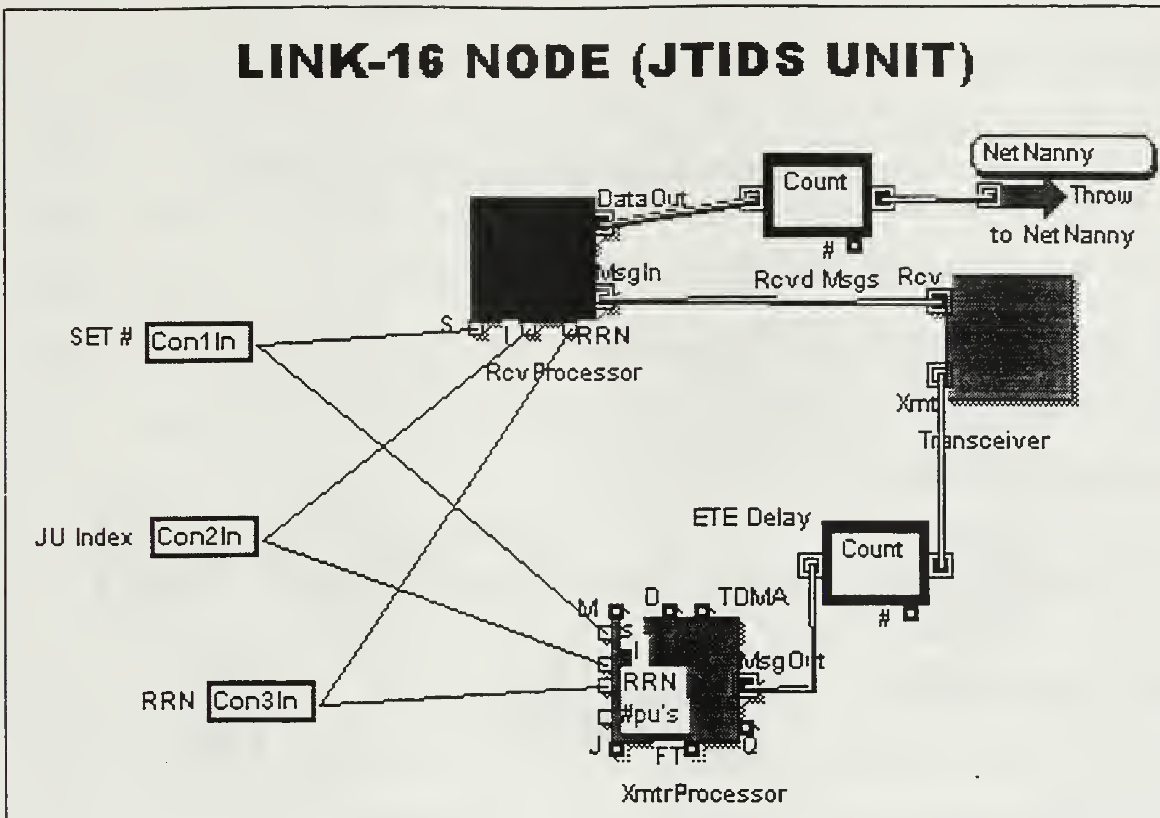

Figure 5-24. JU module.

The Transmitter Processor (Figure 5-25) contains two message generators, one to generate fixed format J-series messages, the other to generate free-text messages. In this program the user can select the distribution used in the message generators as well as the message arrival interval (seconds) and message size. This provides flexibility to evaluate different loads. All messages (items) are tagged with the JU's attributes "Set," RRN, and index number. The end-to-end (ETE) Latency block (Figure 5-26) reads the link

parameters and calculates a message delay. In communications Mode 1, standard data packing mode, 545 bits can be transmitted in each allotted time slot. When transmitting a fixed format (J-series) message, the time slot will contain 210 bits of effective data; the remainder is error encoding and overhead. To calculate the time delay for this TDMA system, first the number of time slots required is determined.

Time Slots = integer ((Message Size bits/210) +1)

Note, the number of slots is rounded up to the next integer value. The message latency can now be calculated from the rate redundancy assigned to the unit and the standard JTIDS time slot arraignment (one time slot per set, every .023438 seconds).

Time Delay (seconds) = .023438 * (2 ** (15 - RRN))

This delay assumes the worst case in that the message must wait a minimum of the time between assigned time slots before it can be transmitted. The calculated delay is forwarded to a time delay block, which holds the outgoing message for the designated time. See Chapter III, System Architecture, for a complete description of JTIDS slot assignments.

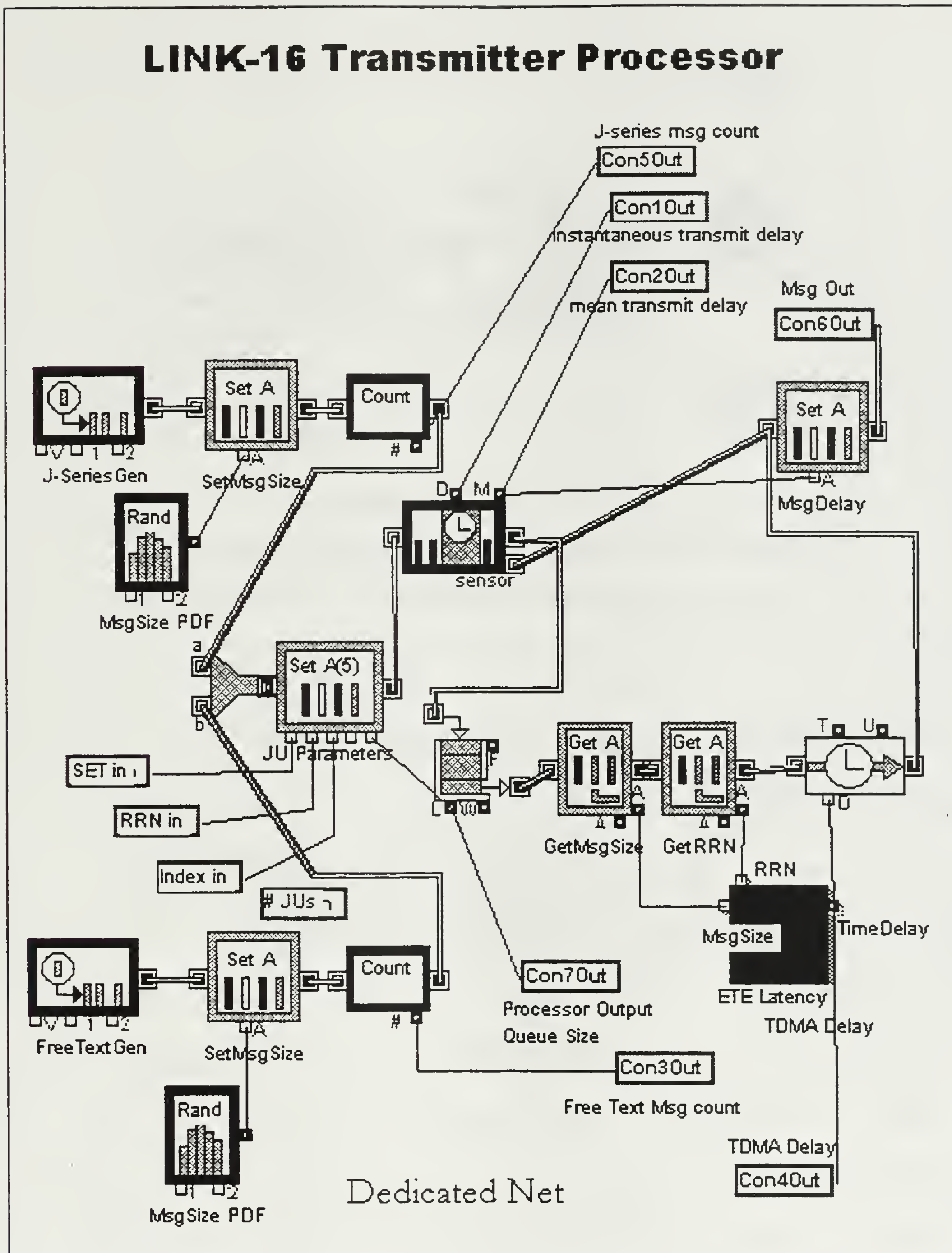

Figure 5-25. Transmitter Processor.

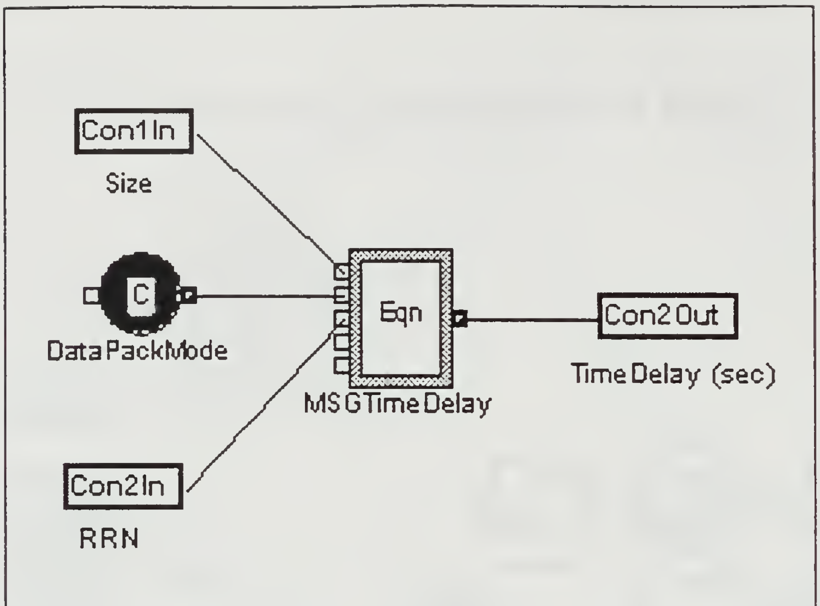

Figure 5-26. End-to-End Latency Block.

The Receiver Processor block (Figure 5-27) compares the received message (item) attributes with the receiver communication parameters to determine if the message is in contention with the units assigned broadcast slots. The incoming message "set" attribute is checked first in the "Check Msg Set" block (Figure 5-28). The message is returned to the broadcast if it is not in the same set as the user. Otherwise, the message proceeds to the "Chk Exclusive" block (Figure 5-29). In the Chk Exclusive block the incoming message link parameters (index, and RRN) are compared to the receiving unit's parameters to determine if

the incoming message is in a time slot mutually exclusive to the receiver's assigned time slots.

Modulus ((index2 - index 1) / (2**(15-RRN1)))

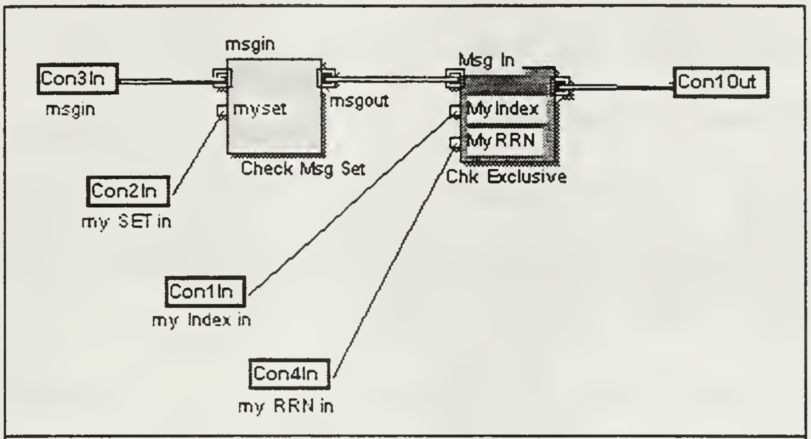

Figure 5-27. Receiver Processor Block.

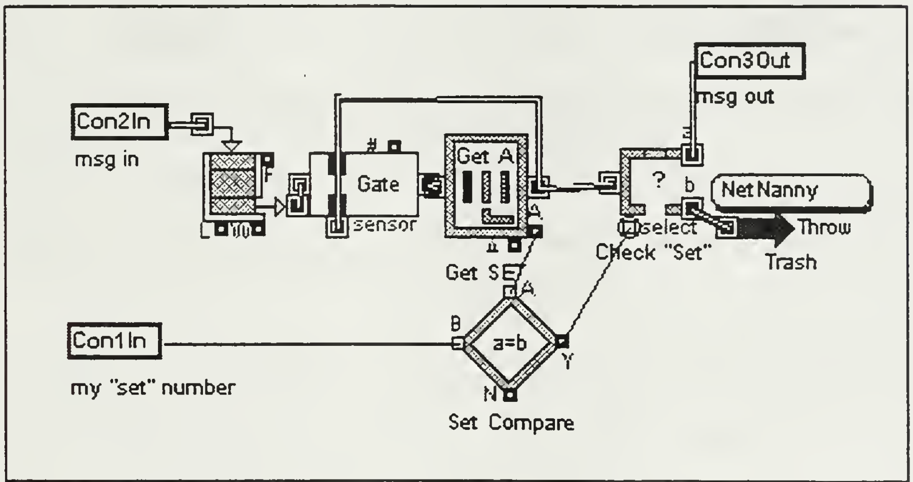

Figure 5-28. Check Msg Set Block.

Index 2 is the larger value. If this returns zero, then the two units are not mutually exclusive. The index of the sending unit is also compared to the receiving unit index. This is necessary because in this model, all units get all messages routed to them by the Broadcast block. If the message is not mutually exclusive (that is it could interfere with the receiving units transmit slots) and it was not sent by the receiving unit (index numbers are different) then the message is counted as an interfering message then sent back to the broadcast. All other messages are counted as received messages and returned to the broadcast. This process can be used to quickly verify a potential system lineup to check for mutual interference. The receiver processor can be used to segregate messages from any time slot assignment. The model could be easily altered to have multiple Receiver Processor blocks to monitor for traffic on other slots simulating a receive-only line-up.

The JTIDS models were developed with an assumption that all units are within 300 nautical miles and within line-of-sight. Another assumption is that the signal strength and robust error correction result in negligible bit errors. Based on these assumptions, the Transceiver block (Figure 5-

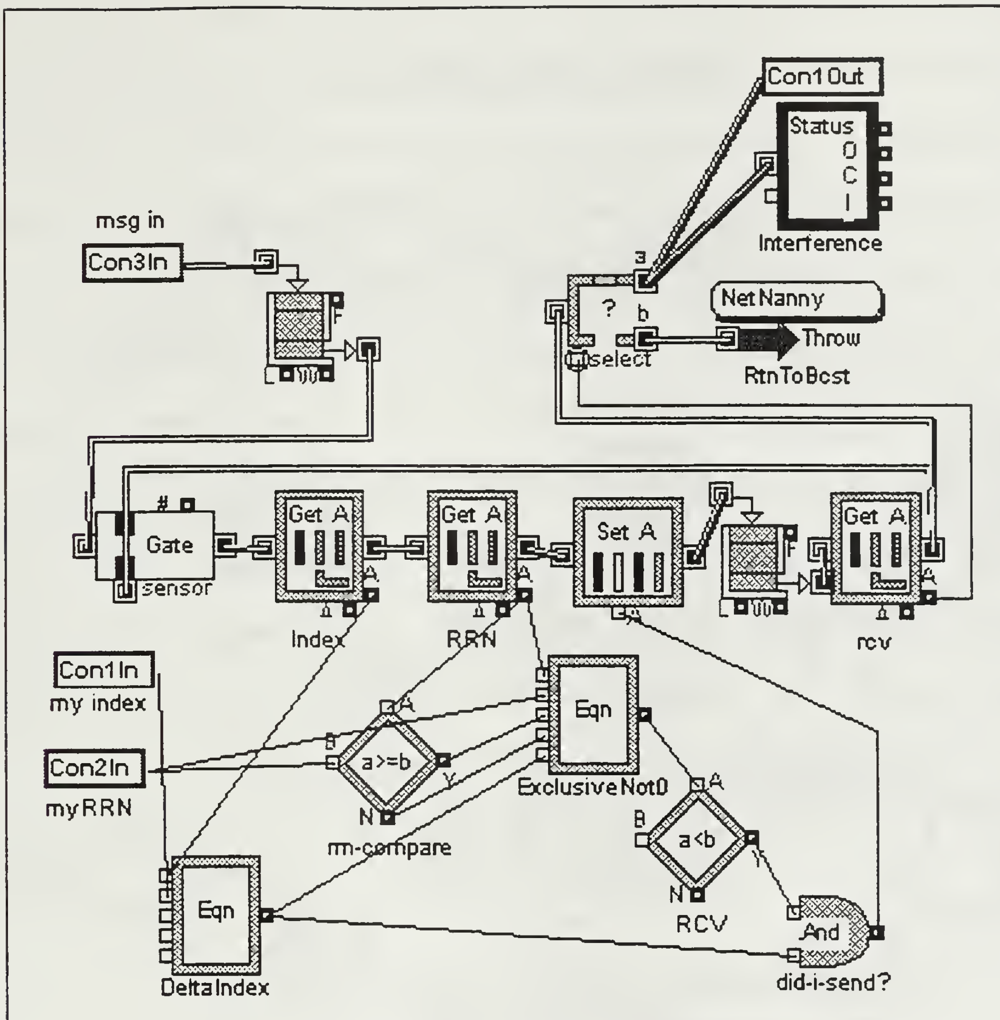

Figure 5-29. Chk Exclusive Block.

30) has been simplified to a buffer (FIFO queue), catch, and throw blocks. The Transceiver block catches or receives and routes messages to the unit's Receiver Processor block or throws outgoing messages to the Broadcast block.

The Broadcast block (Figures 5-31 and 5-32) is used to simulate a radio broadcast. The block receives the message

items transmitted by each of the JTIDS Units, counts them, and then sets a counter attribute equal to the number of net participants. The counter will be used later to remove the message from the broadcast. The message is then forwarded to the first JU, which begins the broadcast cycle. Each JU reads the message and returns it to the Broadcast block. Messages, received back from a unit's Receiver Processor block, are sent to a sorter, which checks and increments the counter then routes the messages to the next unit in the sequence. When all units have seen the message (the counter reaches zero) it is removed from the broadcast and sent to the Net Data block to collect selected data.

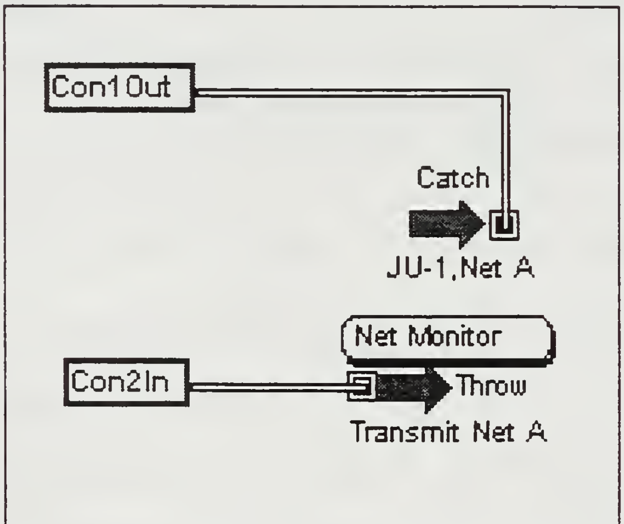

Figure 5-30. Transceiver Block.

Broadcast (1 of 2)

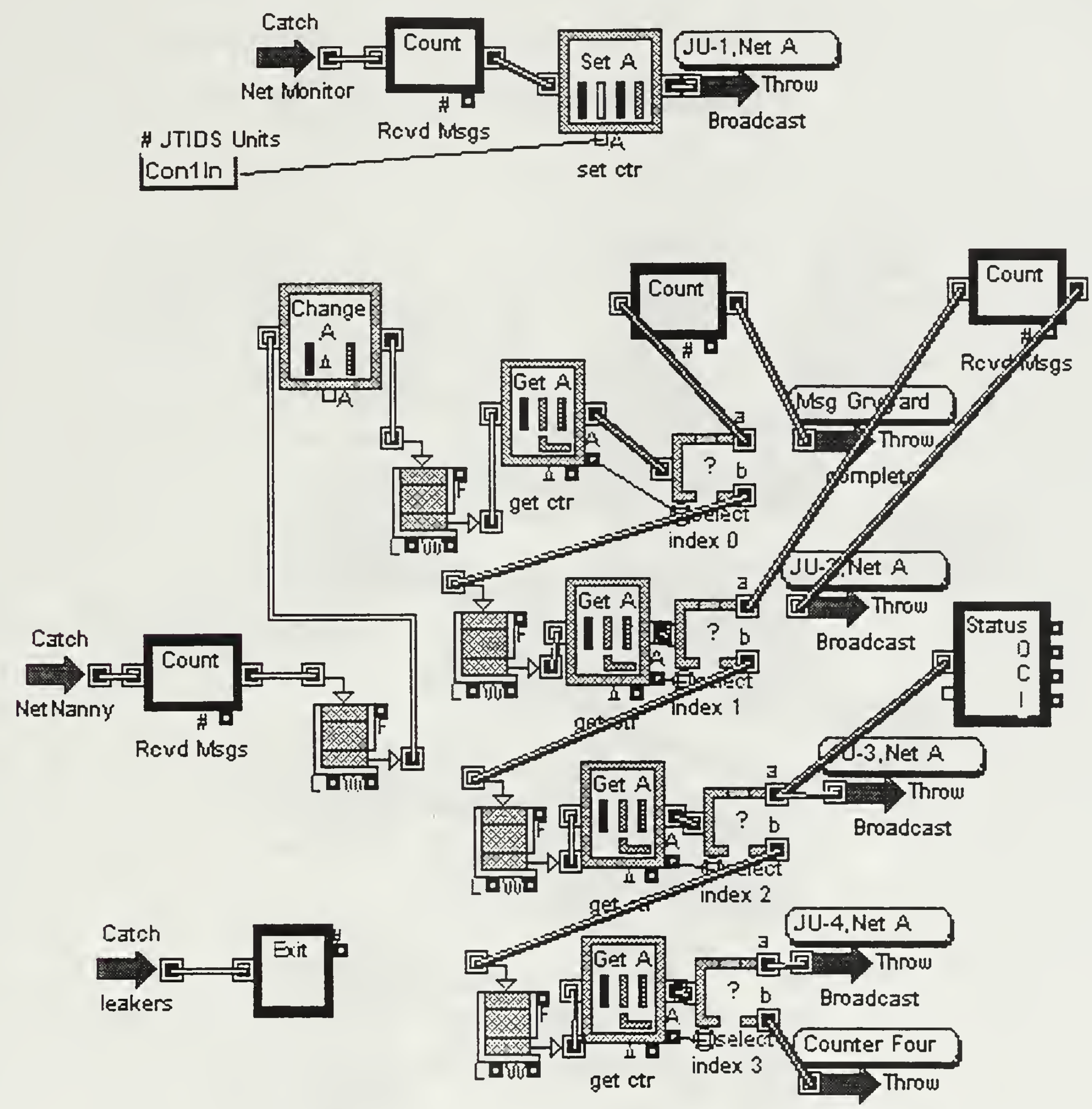

Figure 5-31. Broadcast Block (Part 1 of 2).

Broadcast (2 of 2)

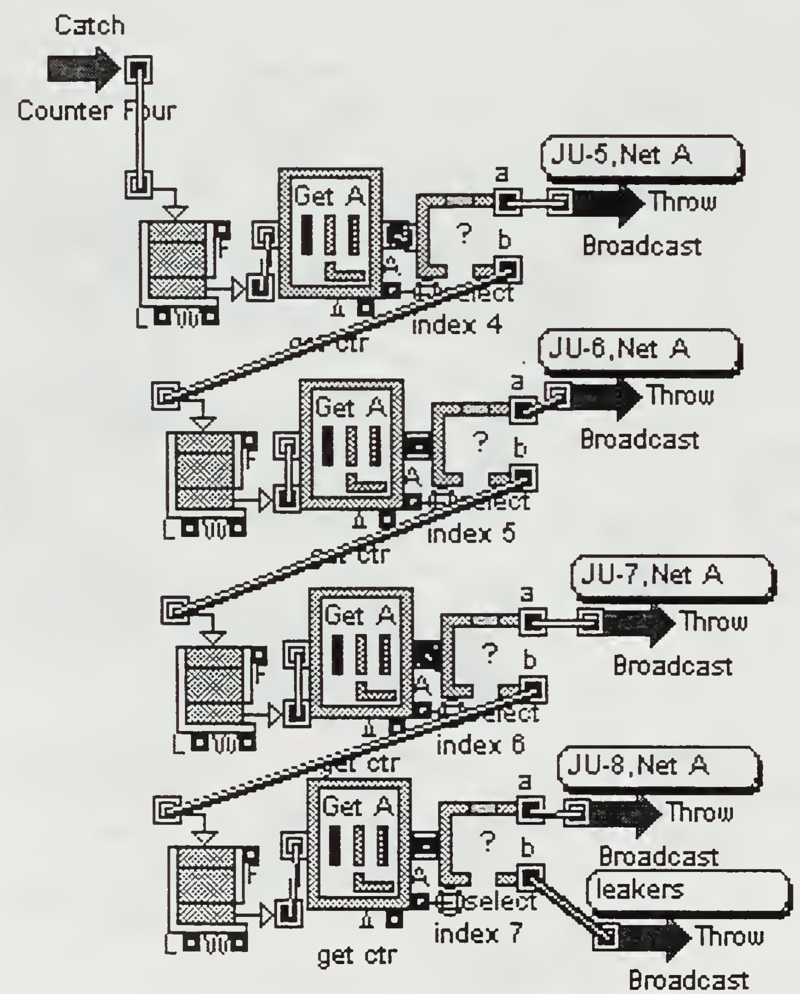

Figure 5-32. Broadcast Block (Part 2 of 2).

The Net Data block (Figure 5-33) is the final stop for all message items. Here the messages are separated by index to plot messages as they are received from each unit. This plot can be compared to the message generation time to see the affects of traffic load on ETE latency.

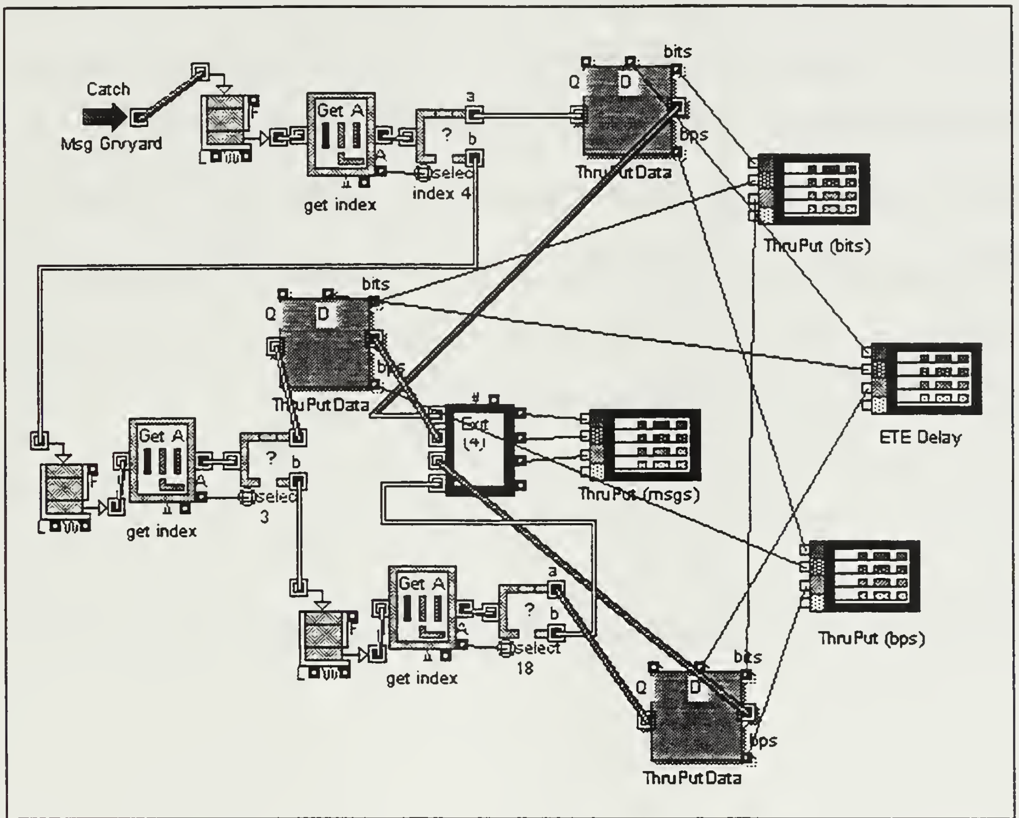

Figure 5-33. Net Data Block.

In the EXTEND model, the network parameters and unit specific time slot assignments are consolidated into an EXTEND notebook (Figures 5-34 and 5-35) to facilitate entering simulation parameters. In the OPNET model the unit

parameters attributes were promoted to the sub-net layer, providing a one stop location to enter the data. Several parameters, such as message generation rate, message size, and distribution, are not accessible to the user. During the model simulation runs the message generation rate was set to 1 message/sec, normally distributed with a 0.5 second standard deviation. The message size was set to a constant 1000 bits/message. The units parameters used for both JTIDS (Link-16) model tests are listed in Table 5-4, JTIDS Slot Group Assignments. Pros and cons of the different models and modeling tools will be discussed in the subjective analysis section of Chapter VII, Conclusion.

Table 5-4. JTIDS Slot Group Assignments.

JTIDS Unit	Net	Slot Group	Set	Index	RRN
1	1	1	C	4	9
2	1	1	C	36	9
3	1	2	B	3	11
4	1	2	B	11	11
5	1	1	C	20	9
6	1	1	C	52	9
7	1	3	C	18	6
8	1	3	C	274	6

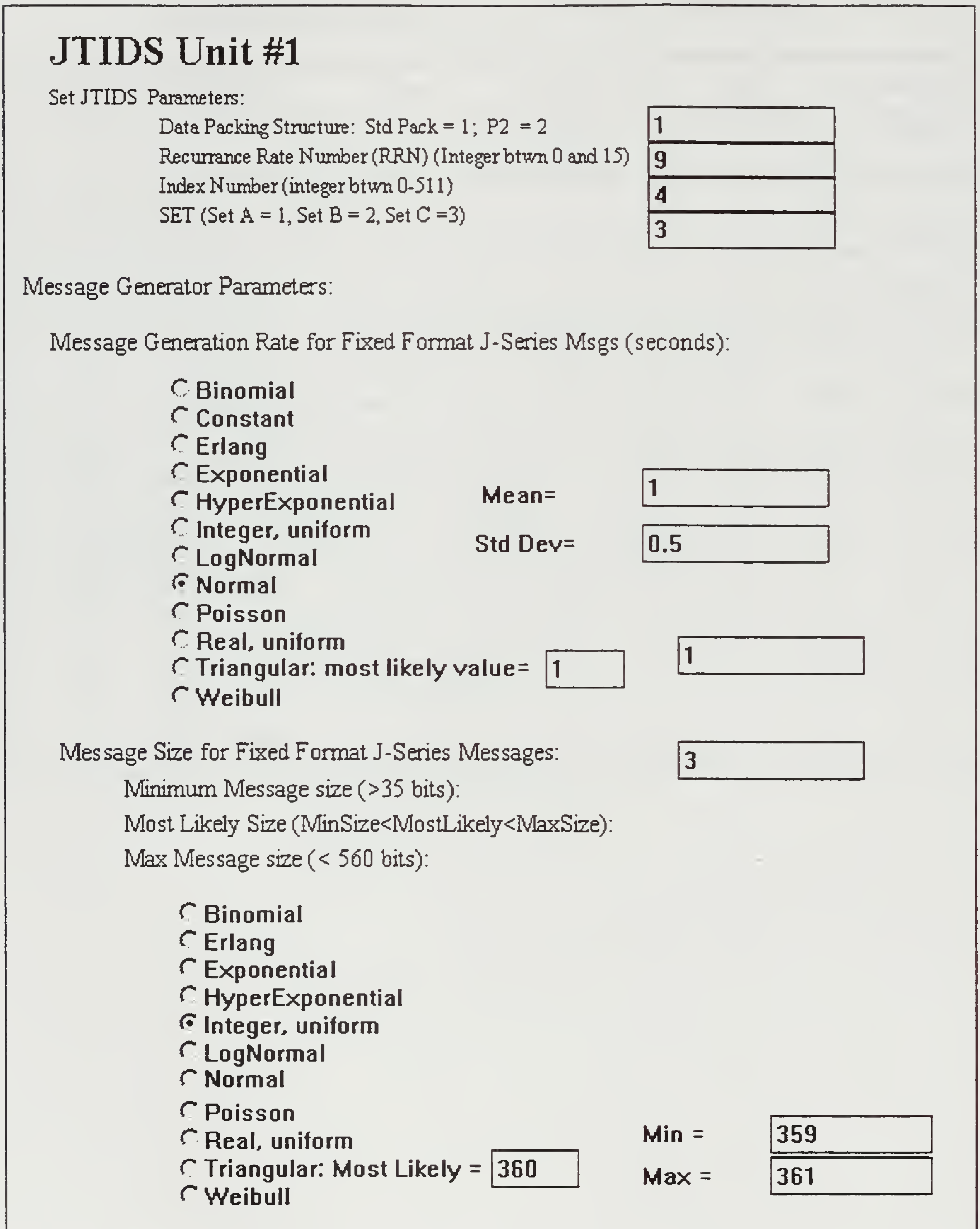

Figure 5-34. EXTEND Notebook (Part 1 of 2).

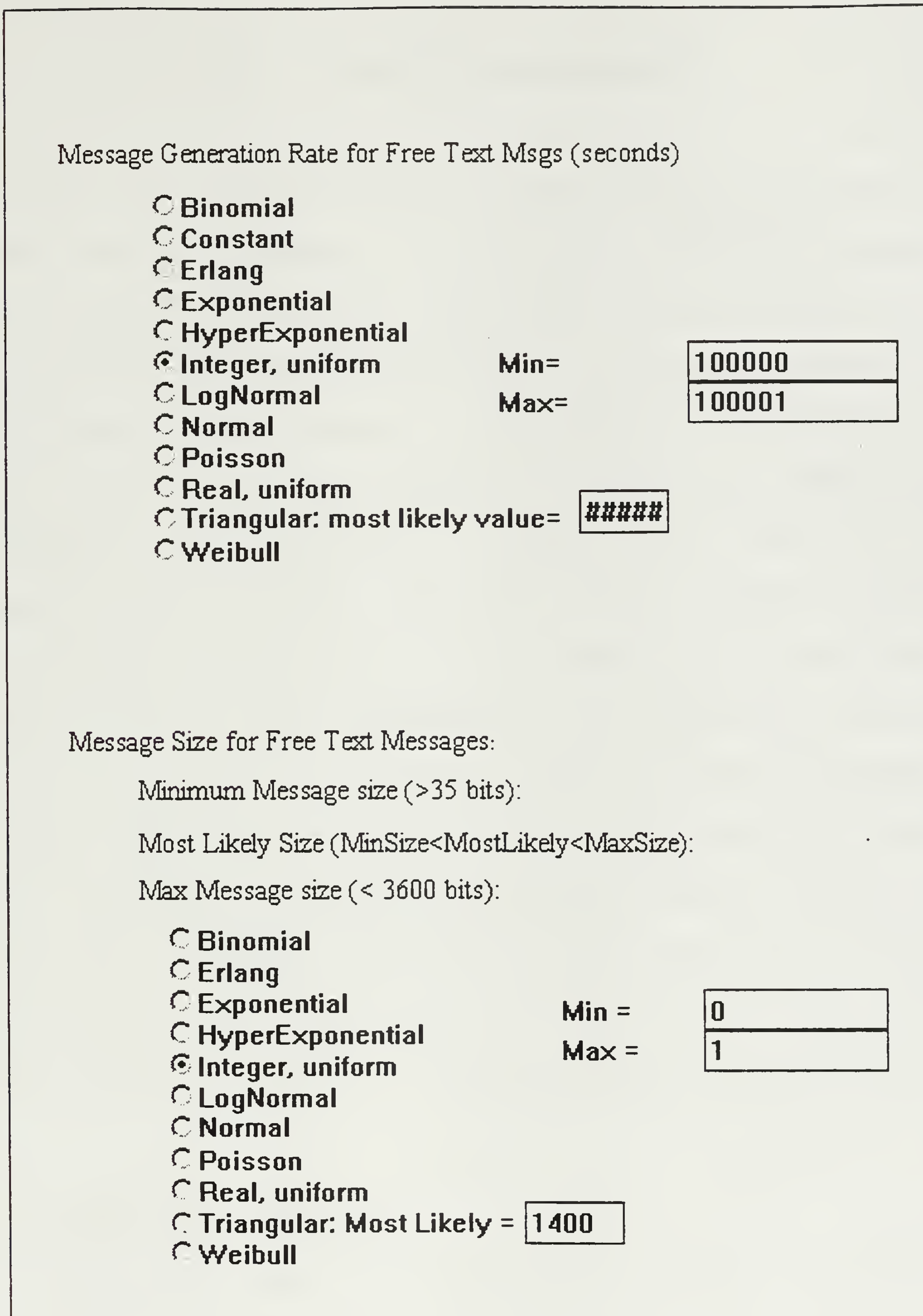

Figure 5-35. EXTEND Notebook (Part 2 of 2).

VI. ANALYSIS OF RESULTS

The overarching purpose of this project is to provide communication planners with the best tools for managing communication systems. To this end, computer models were developed to accomplish two goals. First, to compare the performance of two computer aided modeling and simulation tools and second, to provide a subjective evaluation addressing the utility of using these tools in an operational environment. The goals of this analysis section are to present the results of the simulations in terms that enable a side-by-side comparison of two specific tools and to provide subjective comments regarding OPNET and EXTEND. Network loads are reviewed briefly as a measure to verify that the models are generating sensible results. Next, the performance measures generated by the models are outlined followed by a brief description of the simulation runs. The final section provides the results of the simulation runs.

A. SIMULATION PARAMETERS

The target load from the Ethernet LAN was 2.592 Mbps. The ATM load was set to 0.432 Mbps when the video

teleconference (VTC) station was idle, for a system load of 3.024 Mbps from one sub-net. When activated, the VTC added an additional 24 Mbps from the ATM LAN for a total load of 27.024 Mbps. The IT-21 models had three categories of workstations loading the network. These were E-mail, file transfer (FTP), and video teleconference (VTC). Each E-mail workstation was programmed to generate 7200 messages per hour using a Poisson arrival rate for an average output of 32,000 bps. Each FTP workstation was set to transfer 3600 files per hour, each average 50,000 bytes long for a target load of 400,000 bps per workstation. Message arrival rates were Poisson distributed and message size was normally distributed. The VTC unit generated a constant 24.0 Mbps load when activated. Video frame size and frame rate determines the VTC data rate. The conference interval and conference duration established how often it was activated and how long the VTC periods lasted. For the analysis, the VTC unit was considered off or on. Performance measures were recorded for each condition.

The JTIDS model consisted of eight JTIDS Units (JUs) operating on three different JTIDS channels or "slot groups." Each unit was programmed to generate a message equivalent to 1000 bits of encoded data, every second. Each slot group had a different number of slots assigned for use.

However, all units within a slot group were assigned the same number of time slots, giving them equal network capacity. Since the units within a group have identical capacity, the results are presented for JU #1, JU #3, and JU #7, which are in Slot Groups One, Two, and Three, respectively.

Both models of a particular "network" were equally loaded to facilitate direct comparison of the results. See Chapter V, Models, for details on the system message generating models.

B. MEASURES

The system performance measures used for comparing the IT-21 models are network load, end-to-end (ETE) message delay, and message throughput. To evaluate the JTIDS models the ETE delay, transmit queue length, and data throughput results are compared. Message delay can be defined in several ways. The intent was to measure the time from message generation (the queuing of a file or message by the source) to the time of complete message reception by the end user. This was not practical in the case of the IT-21 model. Messages were generated for network loading but the

data collection point differed between the two models. The OPNET model measured delay time starting at the workstation application level so that the delay time includes processing the data through the data link layer. The EXTEND model measures the time a packet or cell leaves the workstation to the time it reaches the destination LAN.

In the JTIDS model, both programs measure ETE delay as the time from a message or packet reaches the output buffer (queue) to the time it is transmitted. The parameter "message queue length" is also collected by monitoring the number of messages queued for transmission at a given time.

Message throughput is presented in two forms depending on the statistics probe available in OPNET. Units of "bits per second" are used when available. The video teleconference (VTC) throughput is measured in total bytes. The EXTEND version of the IT-21 model also measures throughput in bits or cells over the simulation period. For these two cases, the total throughput was converted to data rate by dividing the mean throughput by the simulation time spent generating it. In OPNET, data rate, as recorded by the statistics probes, is calculated during the simulation run by dividing the cumulative data throughput by the current simulation time. In the JTIDS models data rate is determined by both models and presented in the results.

Two characteristics of VTC operations were modeled; a constant bit rate generator and time sensitive data delivery. To measure the ability to model these VTC attributes, VTC cell arrival was plotted or VTC cell ETE delay was measured.

C. DATA COLLECTION

Simulation runs for the IT-21 models were 110 seconds initially. The OPNET version of this model completed a single run between 1-2 hours. The EXTEND version required approximately 24 hours. After reviewing several of the runs, the simulation was shortened to 60 seconds of simulation time. The EXTEND model reached a steady state in less than 20 seconds into the run. The shorter simulations completed in 1-2 hours per run. The IT-21 model was run 36 times with OPNET and 10 times with EXTEND. The JTIDS model was run 30 times with each modeling tool. All runs produced very consistent results.

The data was collected from the plots and statistics probes with two notable exceptions. First, total throughput needed to be converted to data rate as discussed above. Secondly, the ETE delay time for ATM cells had to be

manually calculated for the EXTEND model results. This had to do with the way time tags are handled with cloned items. Instead, the plot was transferred to spreadsheets and used to determine when a message packet entered the system and when it's cloned cells arrived at the destination. The message size attribute and approximate time of creation positively identified the clones. Thirty sample points were randomly selected to obtain a mean ETE cell latency. This shortfall was corrected for the JTIDS model. Message generation rate, message size, and their respective distributions affect system load and performance. These values were discussed earlier.

D. RESULTS

1. IT-21 System Models

The results of the IT-21 simulation runs are tabulated in Table 6-1, Data Throughput, IT-21 Model, and Table 6-2, ETE Delays, IT-21 Model. The OPNET modeled throughput was less then expected (Figure 6-1, IT-21 OPNET Model Throughput). This is the result of traffic going to other stations within a network and not flowing through the local ATM links and WAN cross connect where the data probes were

located for throughput. The relative throughput was consistent with that seen in the EXTEND model (Figure 6-2, IT-21 EXTEND Model Typical Workstation Throughput). Both models' throughput responded as expected to a VTC (Figure 6-3, IT-21 OPNET Model Throughput with VTC).

Table 6-1. Data Throughput, IT-21 Model.

Node			**OPNET**	**EXTEND**
ATM LAN (Mbps)	**No VTC**	**Mean**	**0.112**	**0.510**
		Std Dev	7.258x10^{-3}	6.24x10^{-2}
	W/ VTC	**Mean**	**0.9173**	**27.01**
		Std Dev	4.516x10-2	2.79x10-2
Ethernet (Mbps)		**Mean**	**0.6143**	**2.82**
		Std Dev	2.846x10-2	0.109
ATM Cross Connect (Mbps)	**No VTC**	**Mean**	**0.3740**	**3.313**
		Std Dev	8.30x10-3	0.104
	w/ VTC	**Mean**	**1.5304**	**29.74**
		Std Dev	7.28x10-2	8.57x10-2

Both models responded similarly to VTC loads. Both show the VTC cells arriving at a very constant rate as indicated by the constant slope in Figure 6-4, IT-21 OPNET Model VTC Throughput and Figure 6-5, IT-21 EXTEND Model VTC Throughput. In the EXTEND model, the ETE delay, of ATM cells generated by the VTC unit, was relatively constant compared to lower priority sources (Table 6-2, IT-21 Model ETE Delays). The VTC traffic had a more apparent affect on other ATM cells delay times as seen when comparing Figure 6-6, IT-21 EXTEND Model Effects of VTC on Non-VTC Cells, and Figure 6-7, IT-21 OPNET Model ATM Cell ETE Delay with VTC. The change in mean ETE delay for cells generated by an ATM workstation (E-mail) and the VTC unit is shown well in Figure 6-8, IT-21 EXTEND Model ETE Delay with VTC. The start of the VTC is very noticeable at 15 seconds into the simulation. Figure 6-9, IT-21 OPNET Model ATM Cell ETE delay indicates that the OPNET model produced longer ETE delay times (Table 6-2). This is attributed to the different locations of the sensing probes between models, as discussed in Section B of this chapter. Otherwise, the results of ATM cell ETE delay were comparable.

The ETE delay times of the Ethernet packets were noticeably less than the delay times obtained from EXTEND and the delay times for ATM cells in the OPNET model. This

is attributed to the ETE Delay statistic probe, which measures the time elapsed from a packet transmission to the time a response is received. The Ethernet LAN is a star topology (shared media). The hub immediately broadcasts each workstation transmission to all the nodes on the medium. Each Ethernet workstation receiving the packet responds with a data unit to the source. The response time within the hub is very short in comparison to the packets and cells going outside the hub, experiencing multiple switches, segmentation, and reassembly. This discrepancy is consistent with the difference in network throughput discussed earlier.

Table 6-2. ETE Delays, IT-21 Model.

Measure			OPNET	EXTEND
ATM ETE (sec)	**No VTC**	**Mean**	**1.386×10^{-4}**	**$5.940\text{x}10^{-6}$**
		Std Dev	$5.341\text{x}10^{-5}$	$2.953\text{x}10^{-7}$
	w/ VTC	**Mean**	**$2.464\text{x}10^{-4}$**	**$1.524\text{x}10^{-5}$**
		Std Dev	$7.01\text{x}10^{-5}$	$6.095\text{x}10^{-6}$
Ethernet ETE (sec)	**No VTC**	**Mean**	**$5.96\text{x}10^{-6}$**	**$3.741\text{x}10^{-3}$**
		Std Dev	Not Available	$1.12\text{x}10^{-3}$
	w/ VTC	**Mean**	**$5.96\text{x}10^{-6}$**	**$4.450\text{x}10^{-3}$**
		Std Dev	$5.51\text{x}10^{-13}$	$5.20\text{x}10^{-4}$
VTC ETE (sec)		**Mean**	Not Available	**$6.312\text{x}10^{-6}$**
		Std Dev		$1.015\text{x}10^{-8}$

2. JTIDS System Models

The two JTIDS Models produced very similar results. Tables 6-3, Data Throughput, JTIDS Model, 6-4, Queue Length,

JTIDS Model, and 6-5, Message Delay Time, JTIDS Model, present the tabulated results.

Table 6-3. Data Throughput, JTIDS Model.

JTIDS Unit		OPNET	EXTEND
JU #1	**Mean**	**314.867**	**333.0**
(bps)	Std Dev	1.72	2.7
JU #3	**Mean**	**1047.1**	**997.6**
(bps)	Std Dev	6.20	12.52
JU #7	**Mean**	**39.71**	**39.92**
(bps)	Std Dev	0.15	3.22

All measures were within 3.5% of one another except JTIDS Unit Three mean queue length. The difference was only by a fraction of a message (less than 200 bits of encoded data) that it is considered negligible. Figures 6-10 and Figure 6-11 compare modeled throughput of one unit from each slot group. Figures 6-12 and 6-13 show the similarity in message queue length for all slot groups. Figure 6-14 and Figure 6-15 reveals the characteristics of JTIDS Unit Three's queue length that was obscured by the scale used in

Figures 6-12 and 6-13. JTIDS Units One and Seven are generating messages at a rate greater than their output capacity. This is indicated by the steady increase in queue length and message delay, as indicated in Figure 6-16 and Figure 6-17. Again the characteristics of JTIDS Unit Three are obscured when plotted with units from the other slot groups. Figure 6-18 and Figure 6-19 is scaled to display the delay time behavior for JTIDS Unit Three. Again, notice the similarity of the two models.

Table 6-4. Queue Length, JTIDS Model.

JTIDS Unit		**OPNET**	**EXTEND**
JU #1	**Mean**	**686**	**662.3**
(msgs)	Std Dev		17.9
JU #3	**Mean**	**0.610**	**0.780**
(msgs)	Std Dev	0.590	0.750
JU #7	**Mean**	**943**	**954.4**
(msgs)	Std Dev		14.9

Table 6-5. Message Delay Time, JTIDS Model.

JTIDS Unit		OPNET	EXTEND
JU #1 (sec)	Max Delay	685.86	662.1
			8.48
JU #3 (sec)	Mean	0.5984	0.570
	Std Dev	0.1562	0.661
JU #7 (sec)	Max Delay	934.4	942.64
			3.27

E. SUMMARY

The data throughput generated with the two IT-21 models differed significantly. The reason for the difference is the way the models route new messages and the locations of the data probes. Both of these discrepancies relate to model design and should be correctable. Perhaps more importantly, both models responded similarly to changes in system load. That indicates perhaps a scaling difference or a difference in system load. The JTIDS model results were

remarkably similar. The design of the EXTEND model allowed the user more flexibility in checking different system loads and provided a more realistic account of usable data by modeling message headers and error encoding. The OPNET version modeled line-of-sight communications using range between units and height of eye to determine if units were over the visual horizon. Bending or ducting was not modeled.

OPNET is definitely more powerful for developing high granularity computer network models. The downside is that OPNET is a very complex modeling tool. The user's manual fills several three-inch binders compared to one paperback book for EXTEND. Despite its complexity, building models with OPNET is fairly straightforward as long as there is a node or process model that models the desired system. Customizing process models or building originals is an extreme jump in complexity. Finally, building models in OPNET and fully understanding the settings and parameters affecting model behavior are two different concerns. With the various layers and levels of complexity, a less than "well informed" user could easily build undesired attributes into a model.

EXTEND is more generic. It's building blocks start at a much lower level. This allows or forces the modeler to

understand the system being modeled and precisely what behavior is or is not modeled. It does not include cartography support or radio propagation models, however if an attribute can be represented with a mathematical model or estimate then it can be modeled with EXTEND. Very large programs, or programs collecting millions of data points can use a lot of system resources. Smaller models run quite fast. The graphics are simple but provide a good visualization of the model. EXTEND supports distributed model development. Systems or functions can be subdivided into component blocks and archived for future use. With EXTEND, the blocks can be developed separately then assembled to form the system model. This could be useful to support operations in a forward area. With a phone line, connections could be provided to the rear area expertise to build the node or object and return the results.

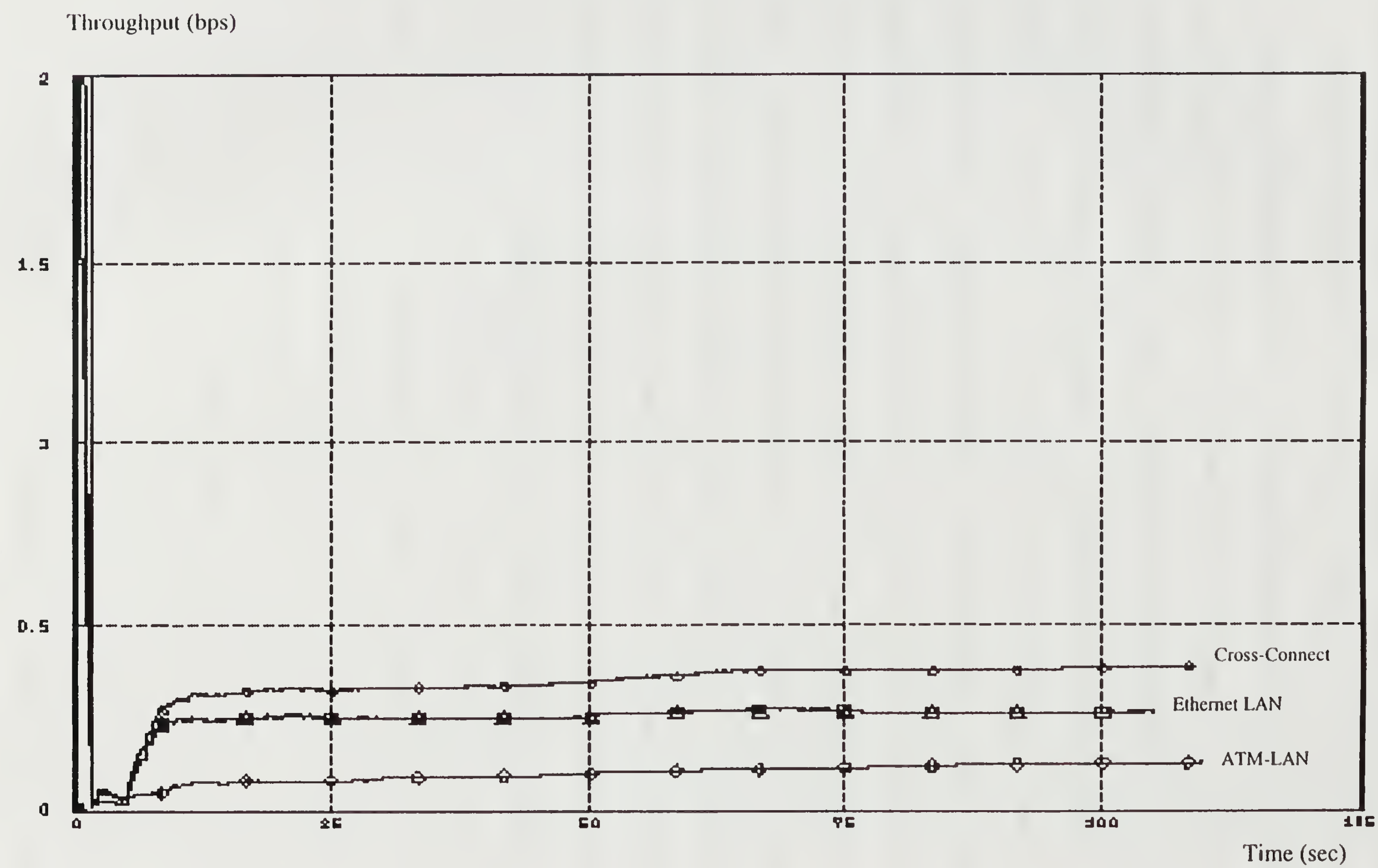

Figure 6-1. IT-21 OPNET Model Throughput.

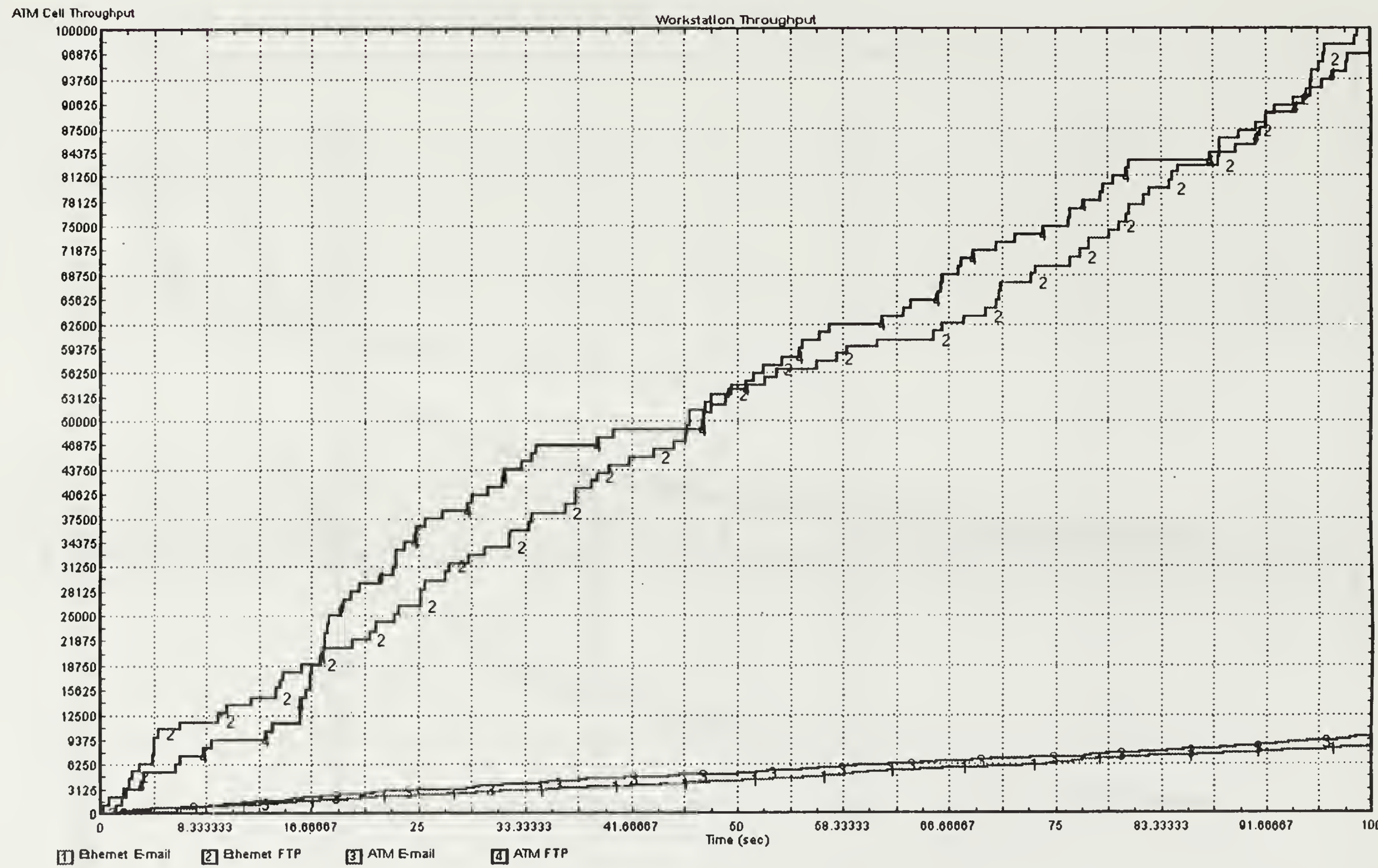

Figure 6-2. IT-21 EXTEND Model Typical Workstation Throughput.

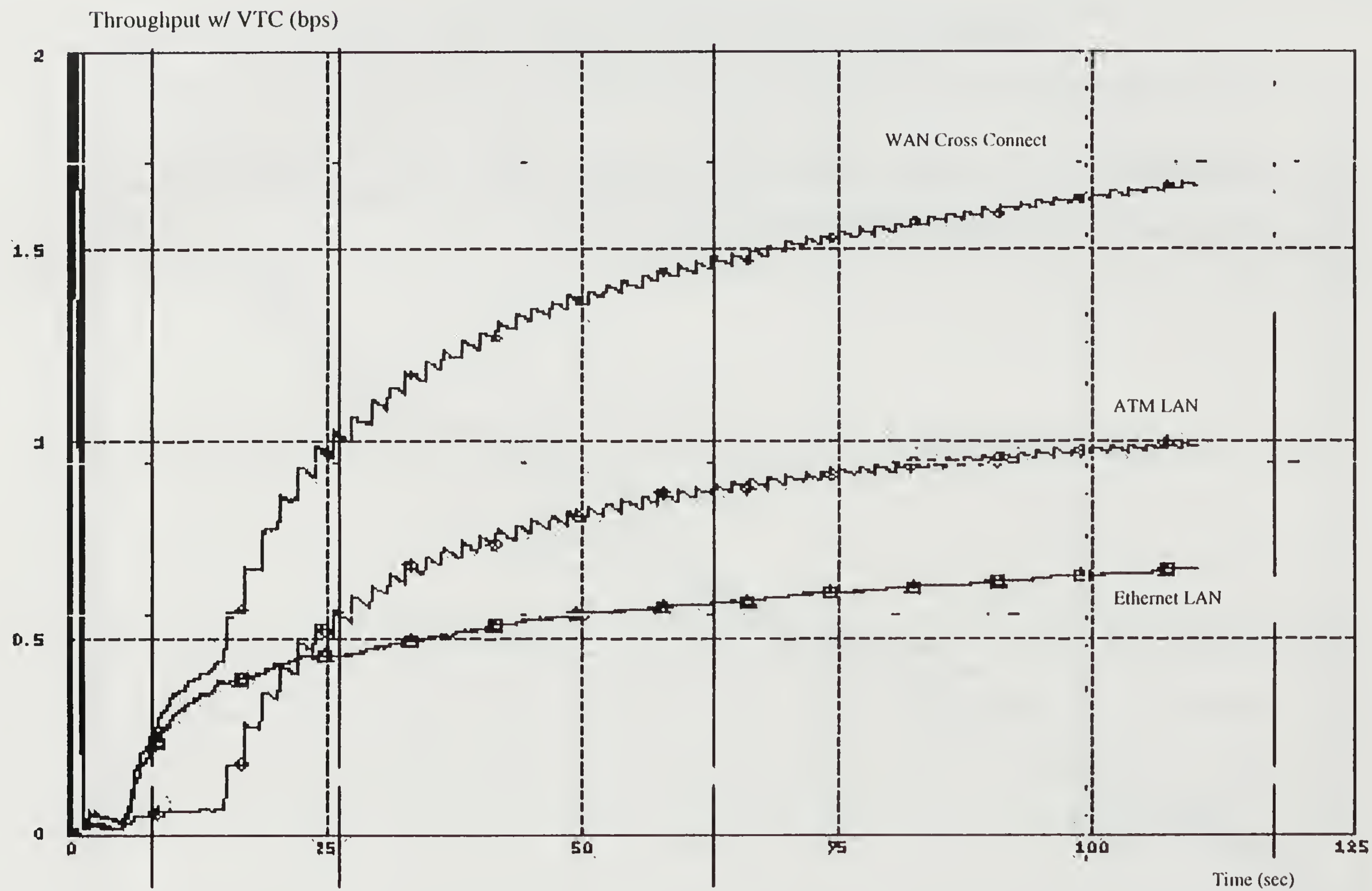

Figure 6-3. IT-21 OPNET Model Throughput with VTC.

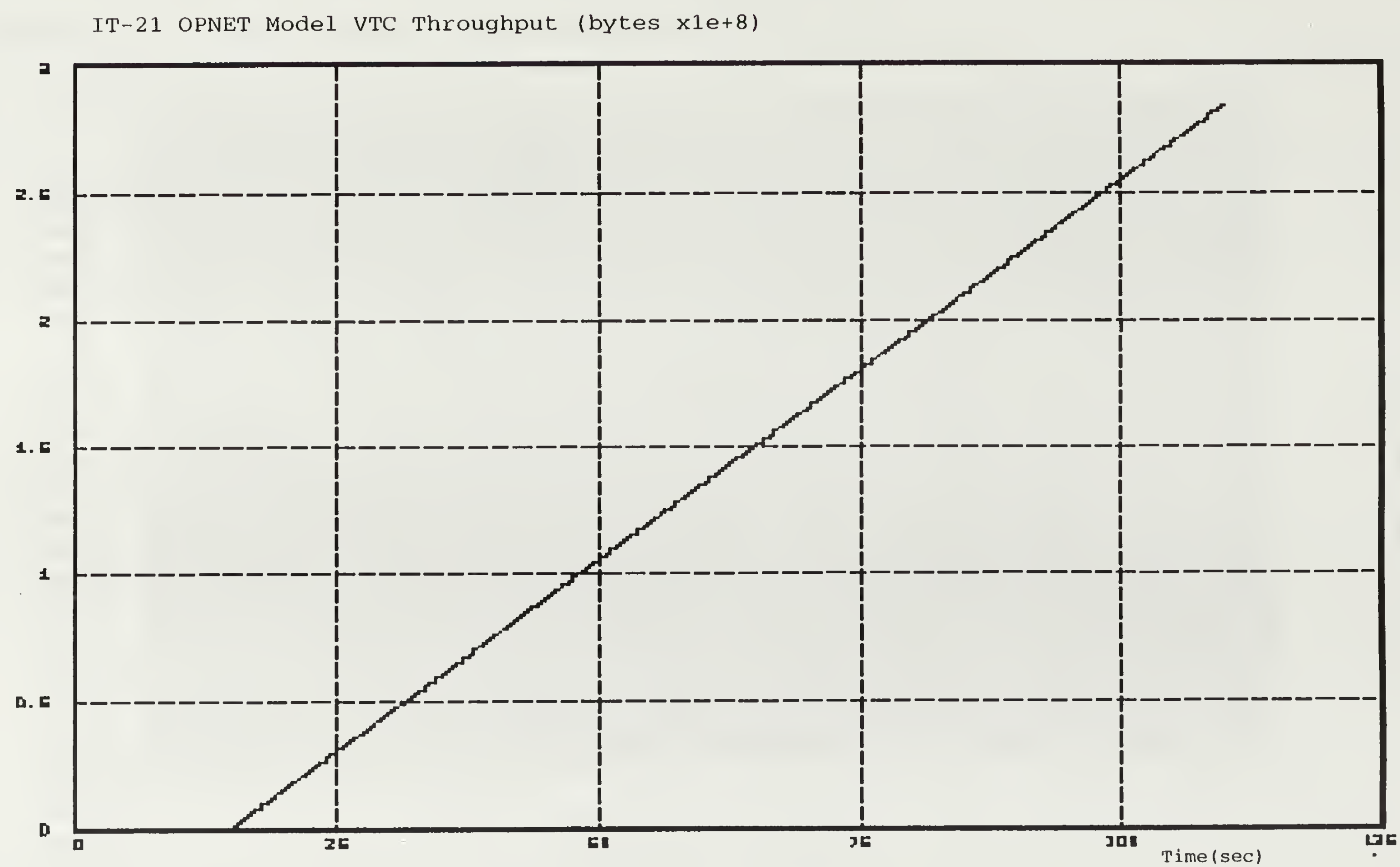

Figure 6-4. IT-21 OPNET Model VTC Throughput.

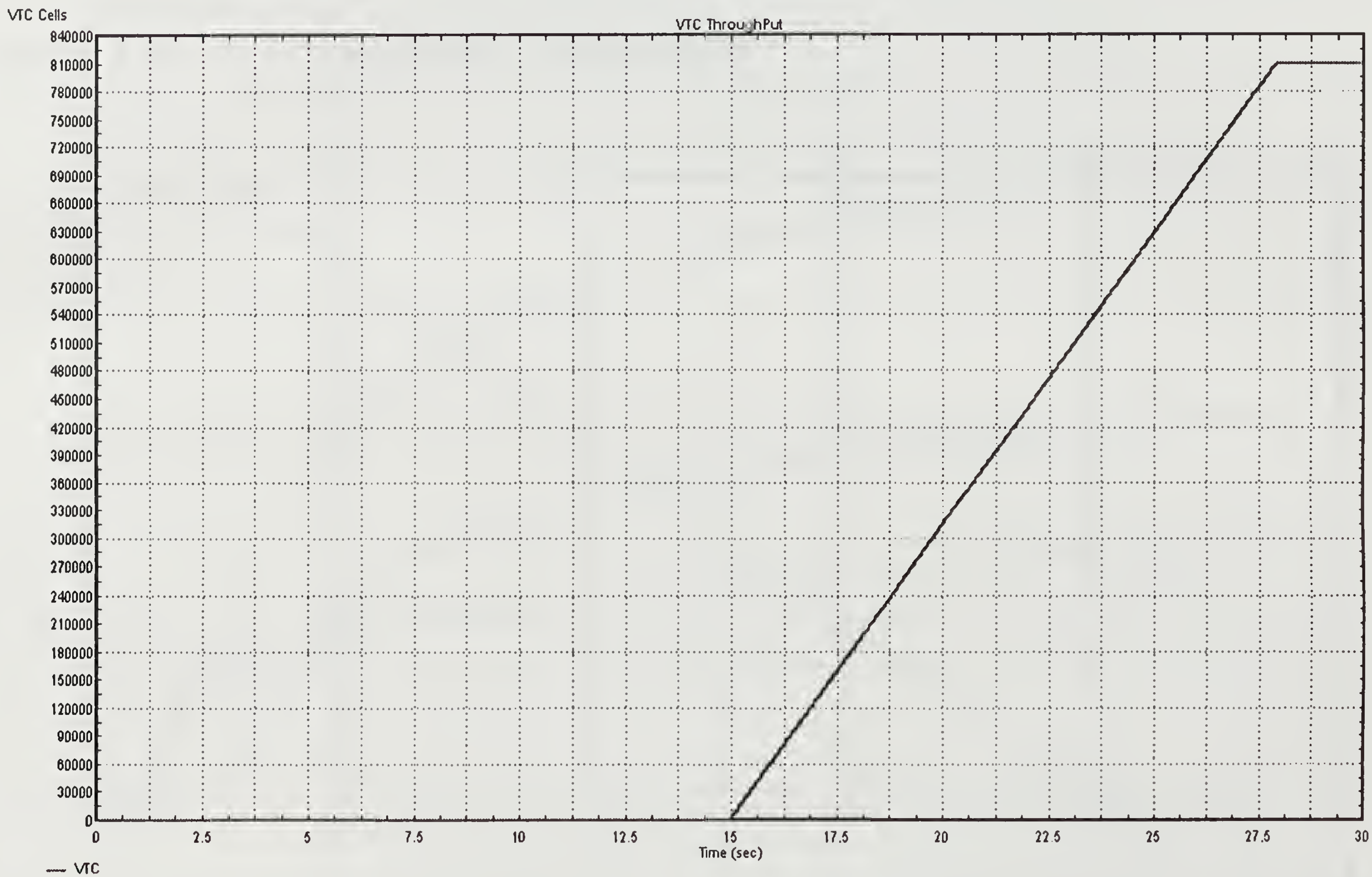

Figure 6-5. IT-21 EXTEND Model VTC Throughput.

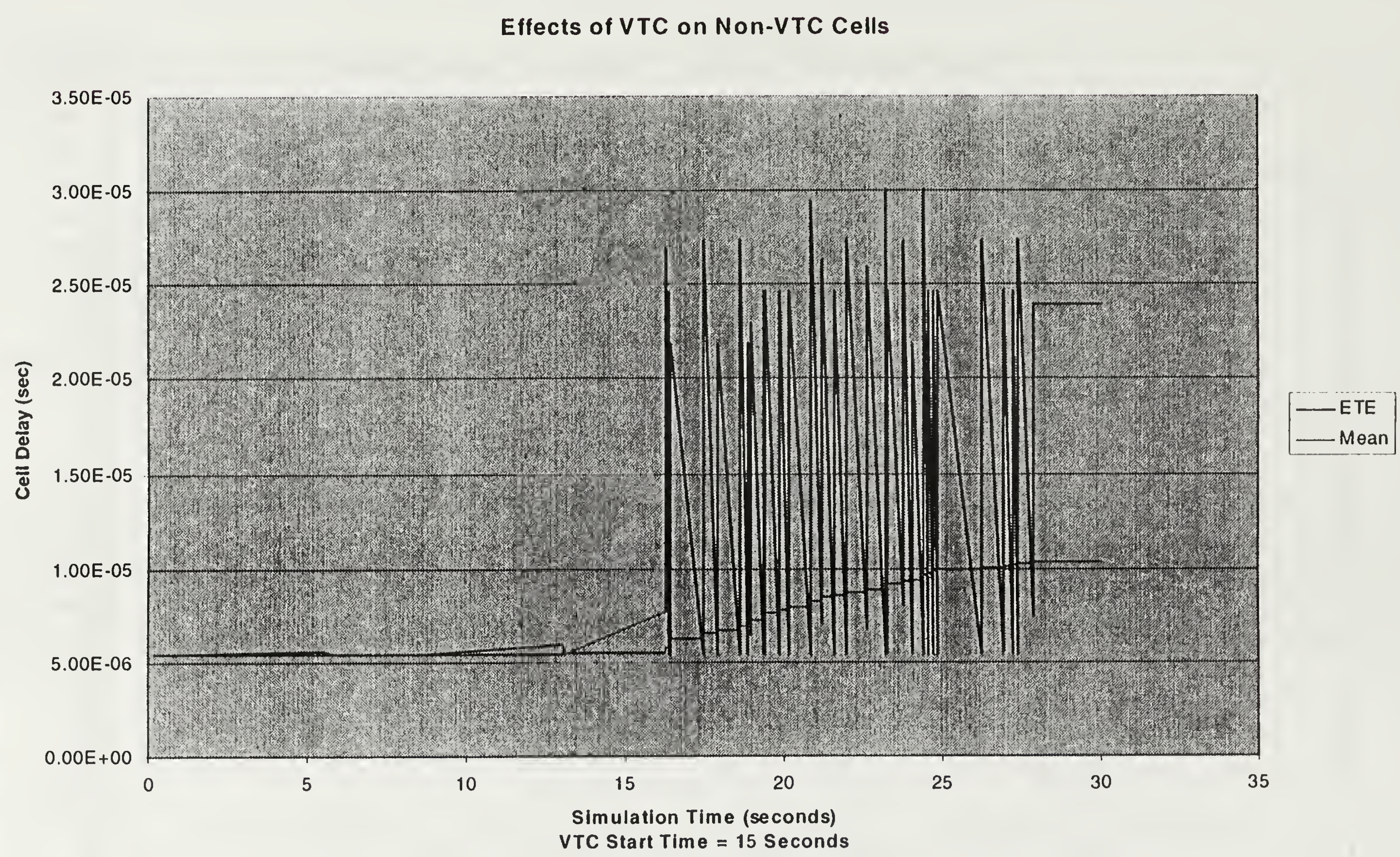

Figure 6-6. IT-21 EXTEND Model Effects of VTC on Non-VTC Cells.

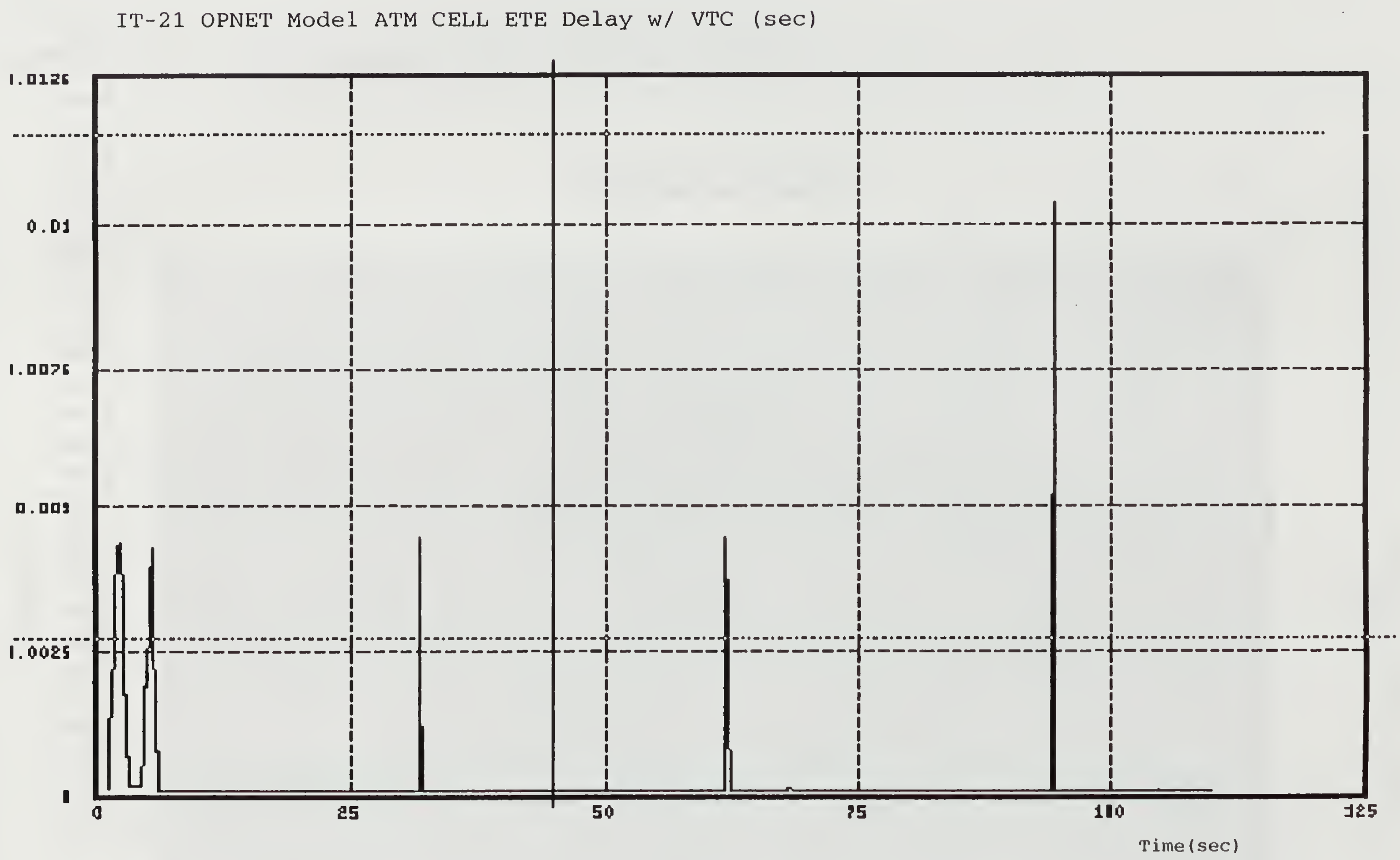

Figure 6-7. IT-21 OPNET Model ATM Cell ETE Delay with VTC.

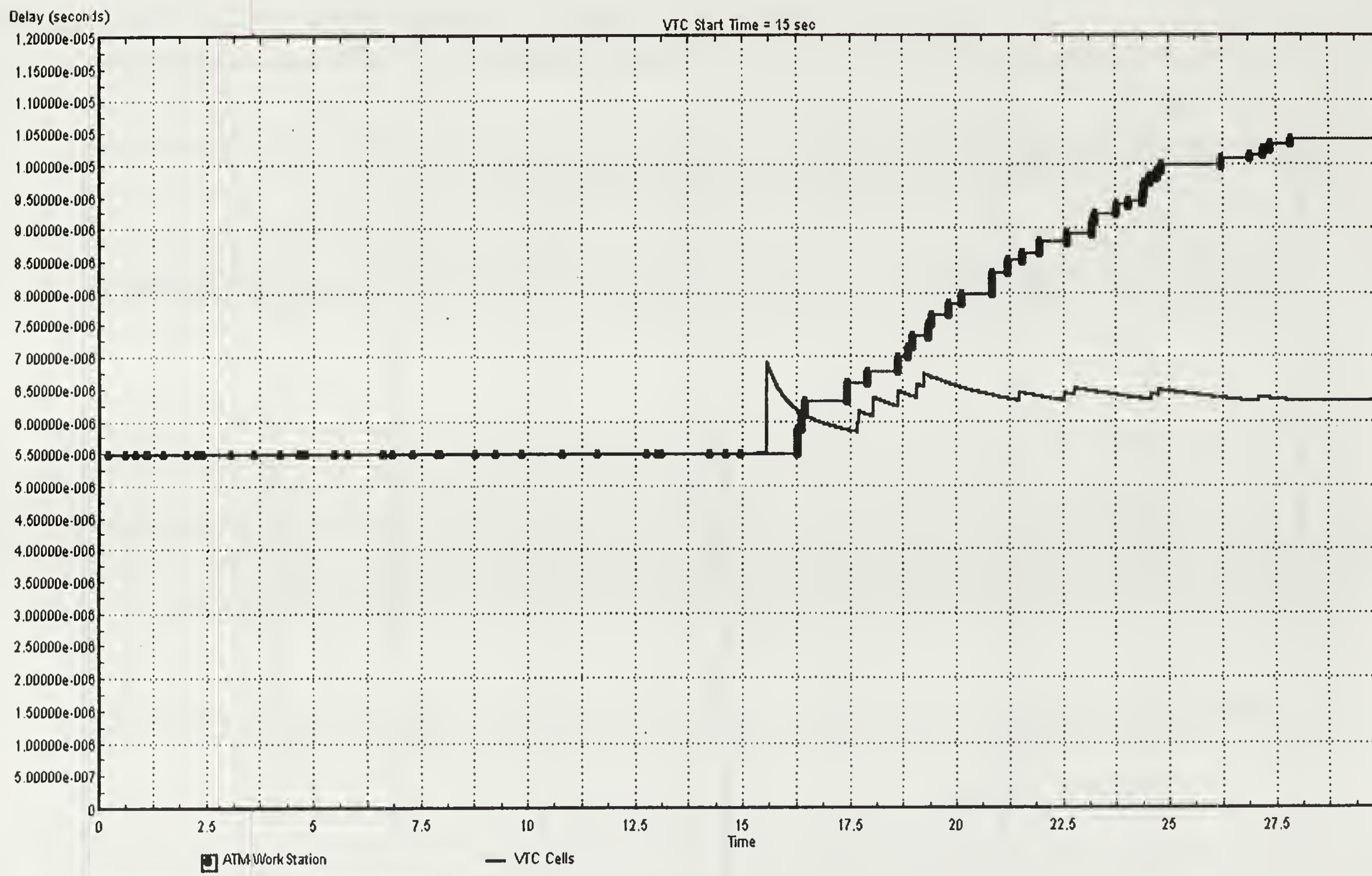

Figure 6-8. IT-21 EXTEND Model ETE Delay with VTC.

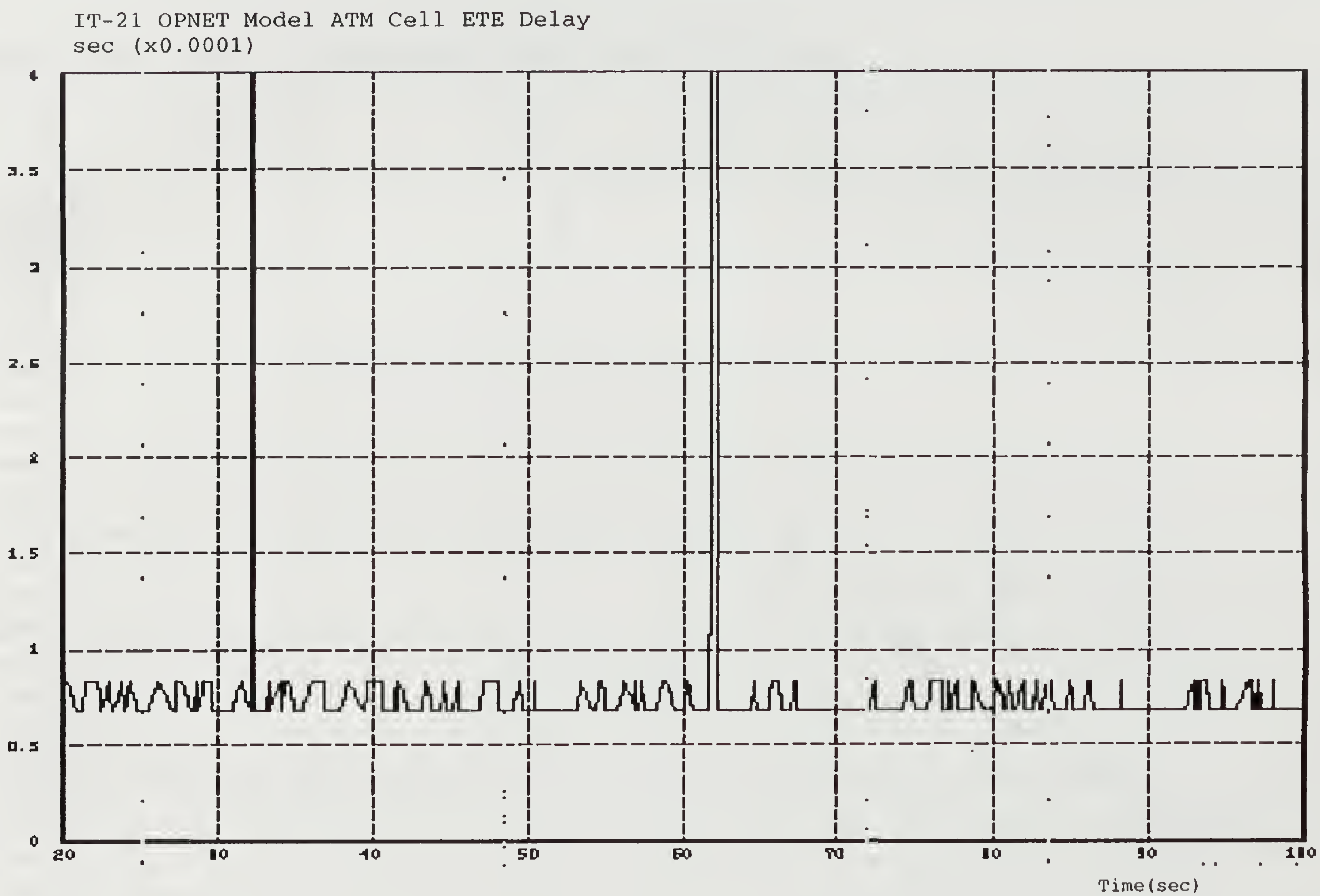

Figure 6-9. IT-21 OPNET Model ATM Cell ETE Delay.

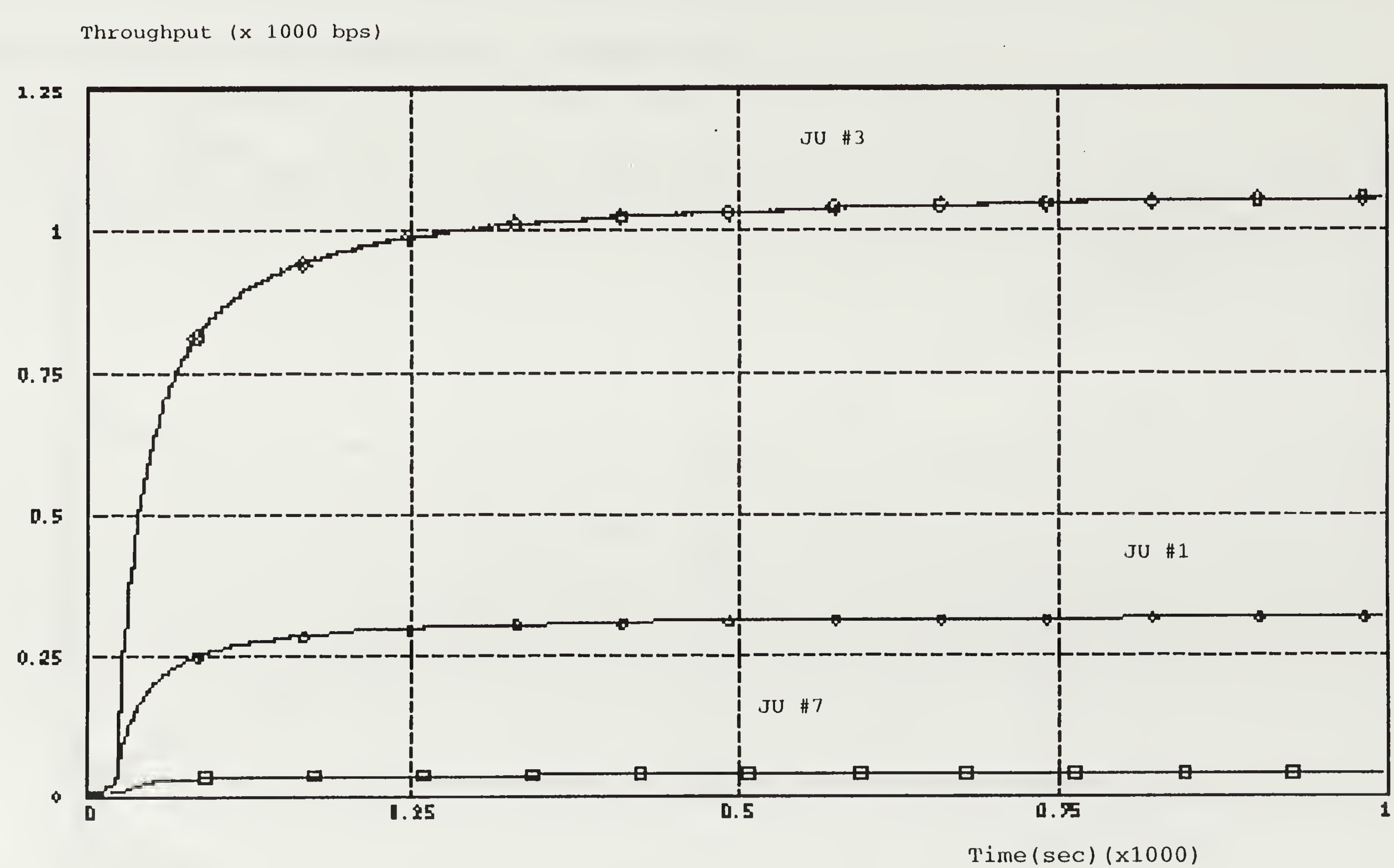

Figure 6-10. JTIDS Throughput, OPNET Model.

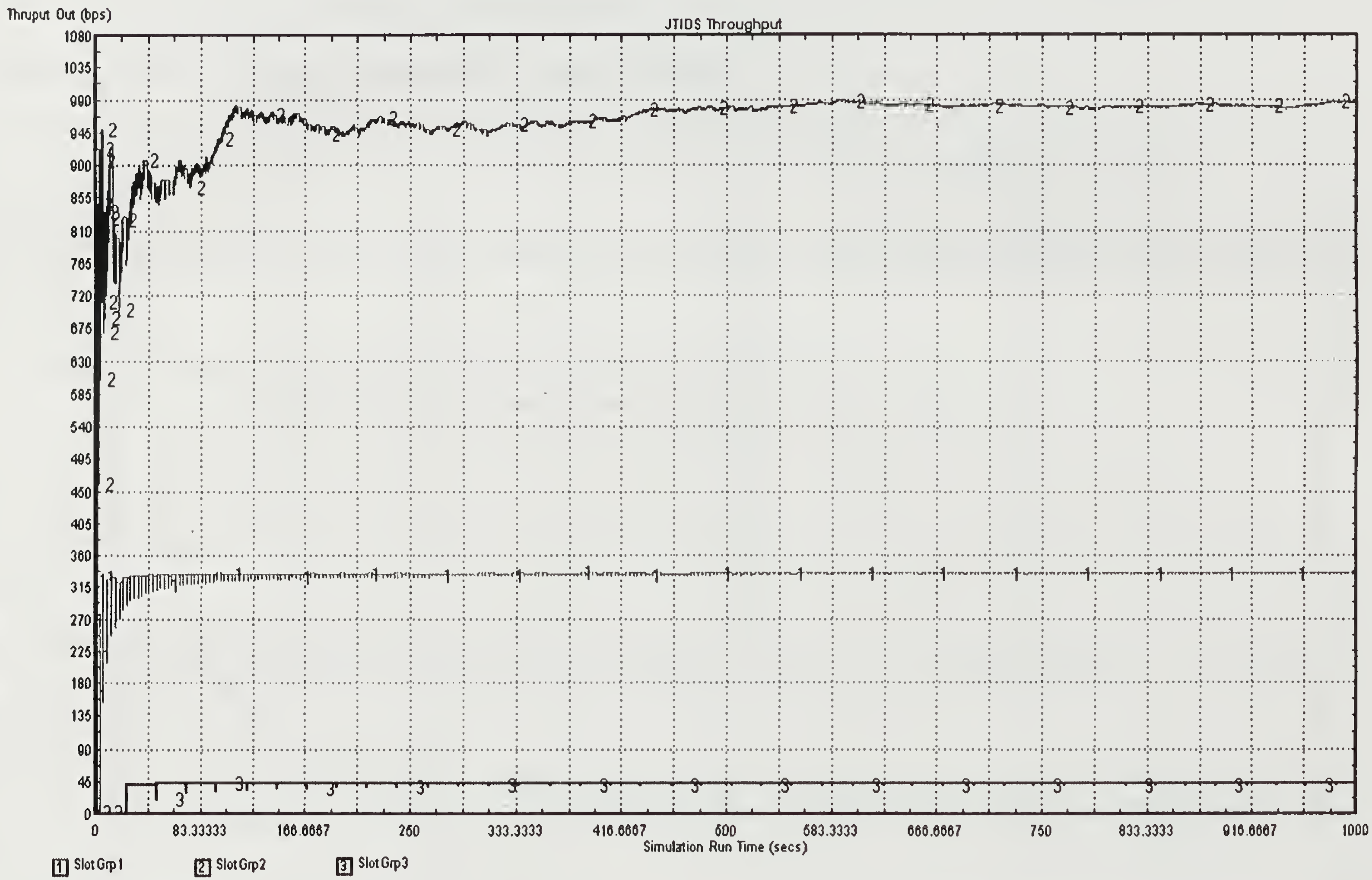

Figure 6-11. JTIDS Throughput, EXTEND Model.

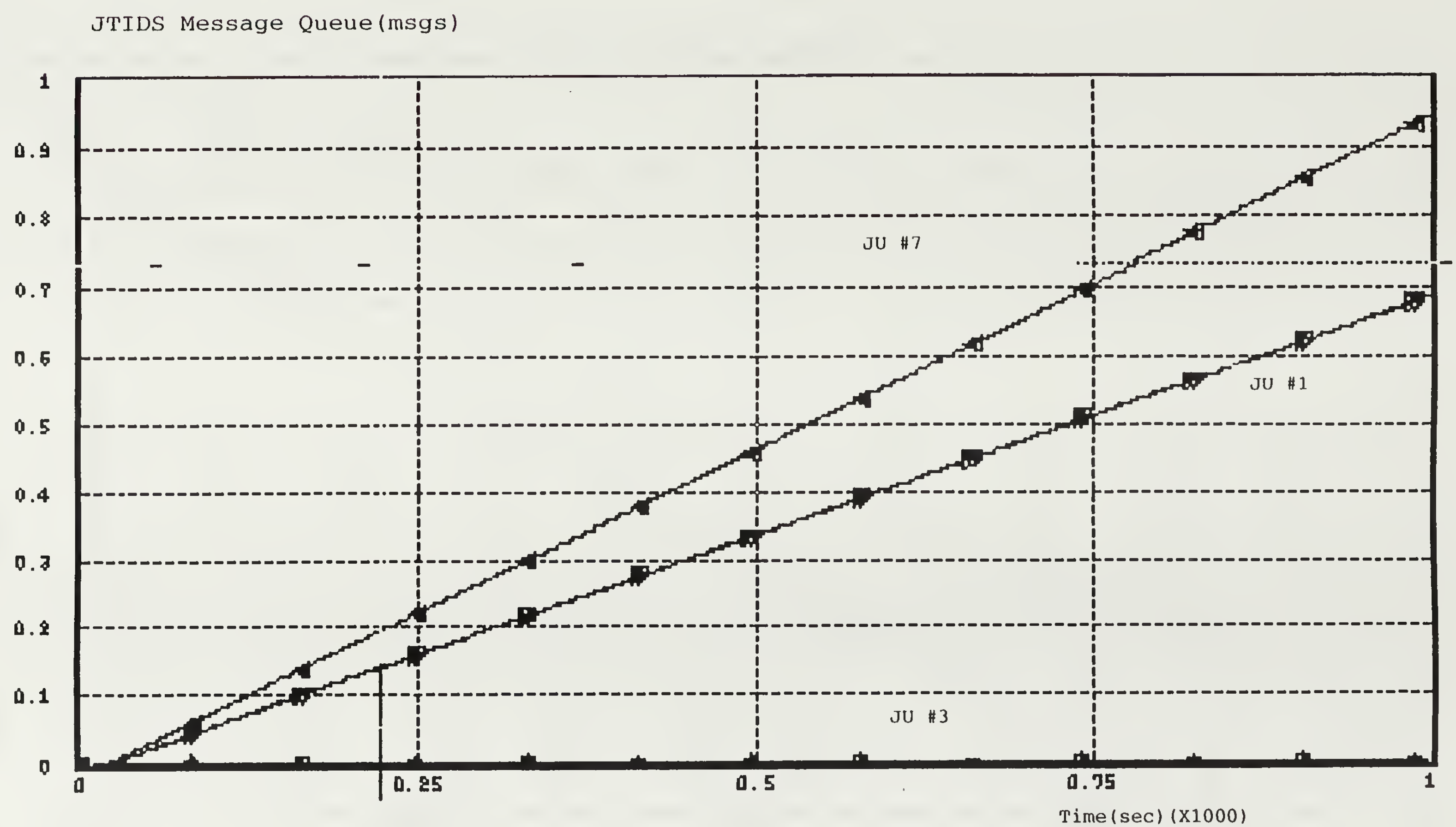

Figure 6-12. JTIDS Message Queue, OPNET Model.

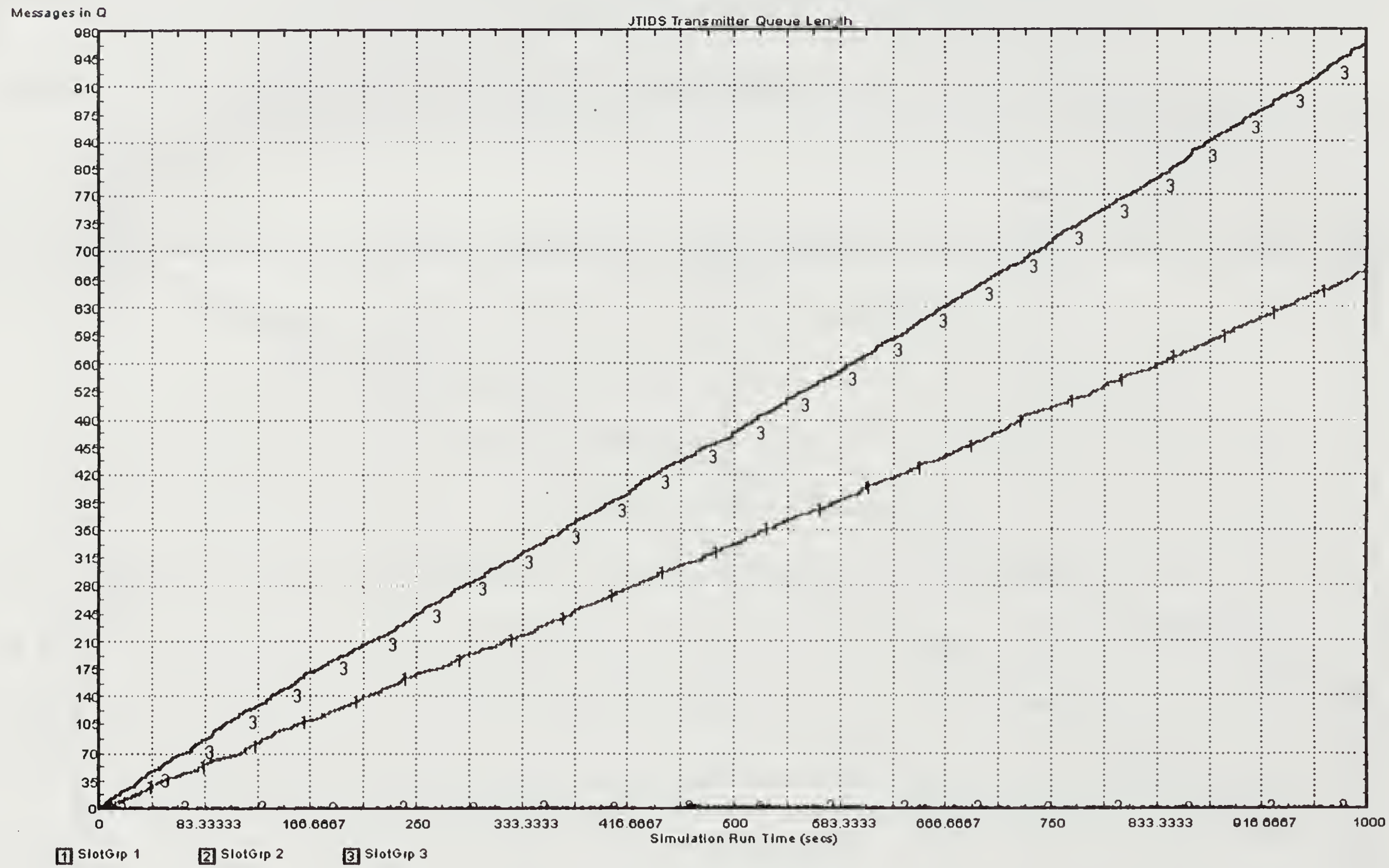

Figure 6-13. JTIDS Message Queue, EXTEND Model.

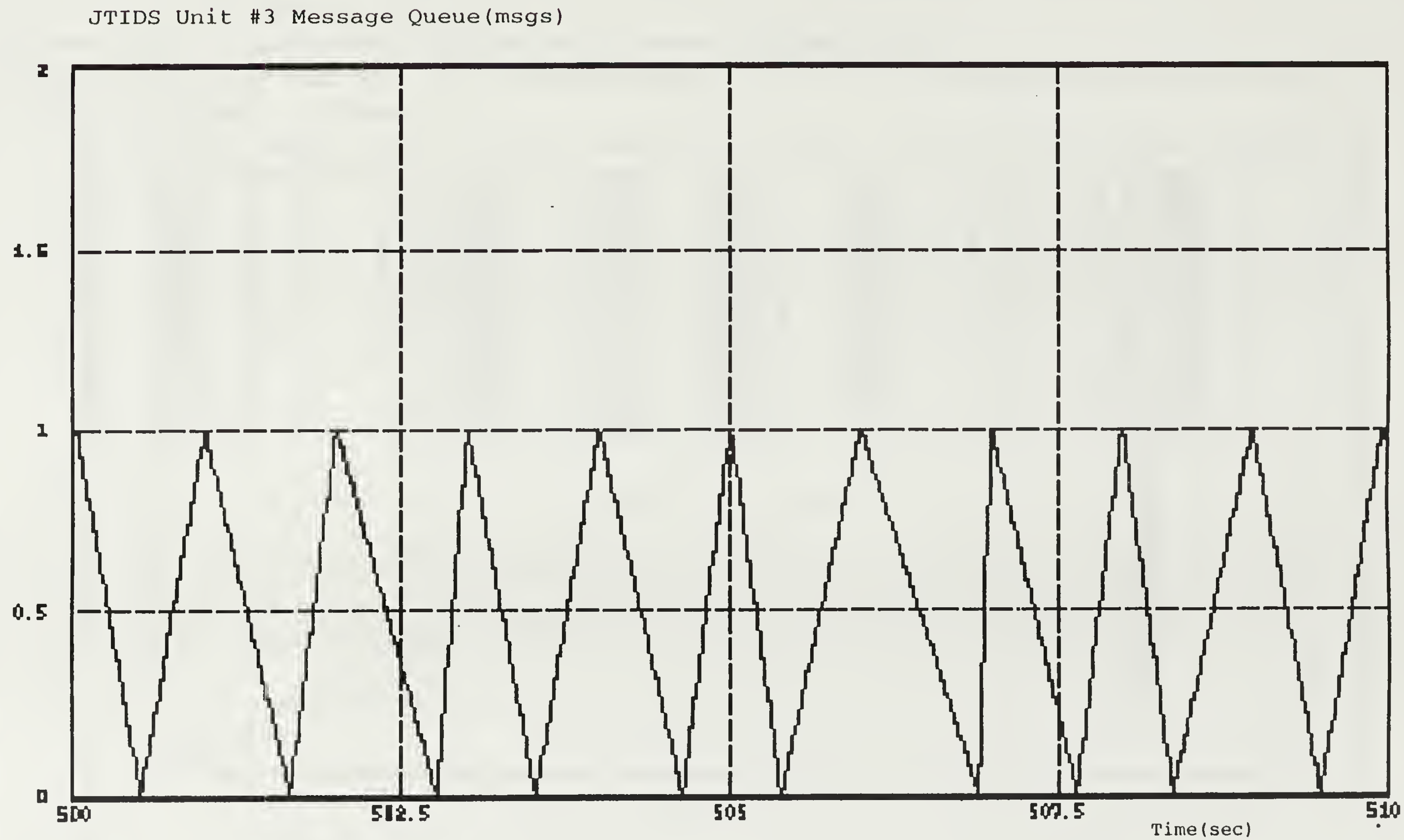

Figure 6-14. JTIDS Unit #3 Message Queue, OPNET Model.

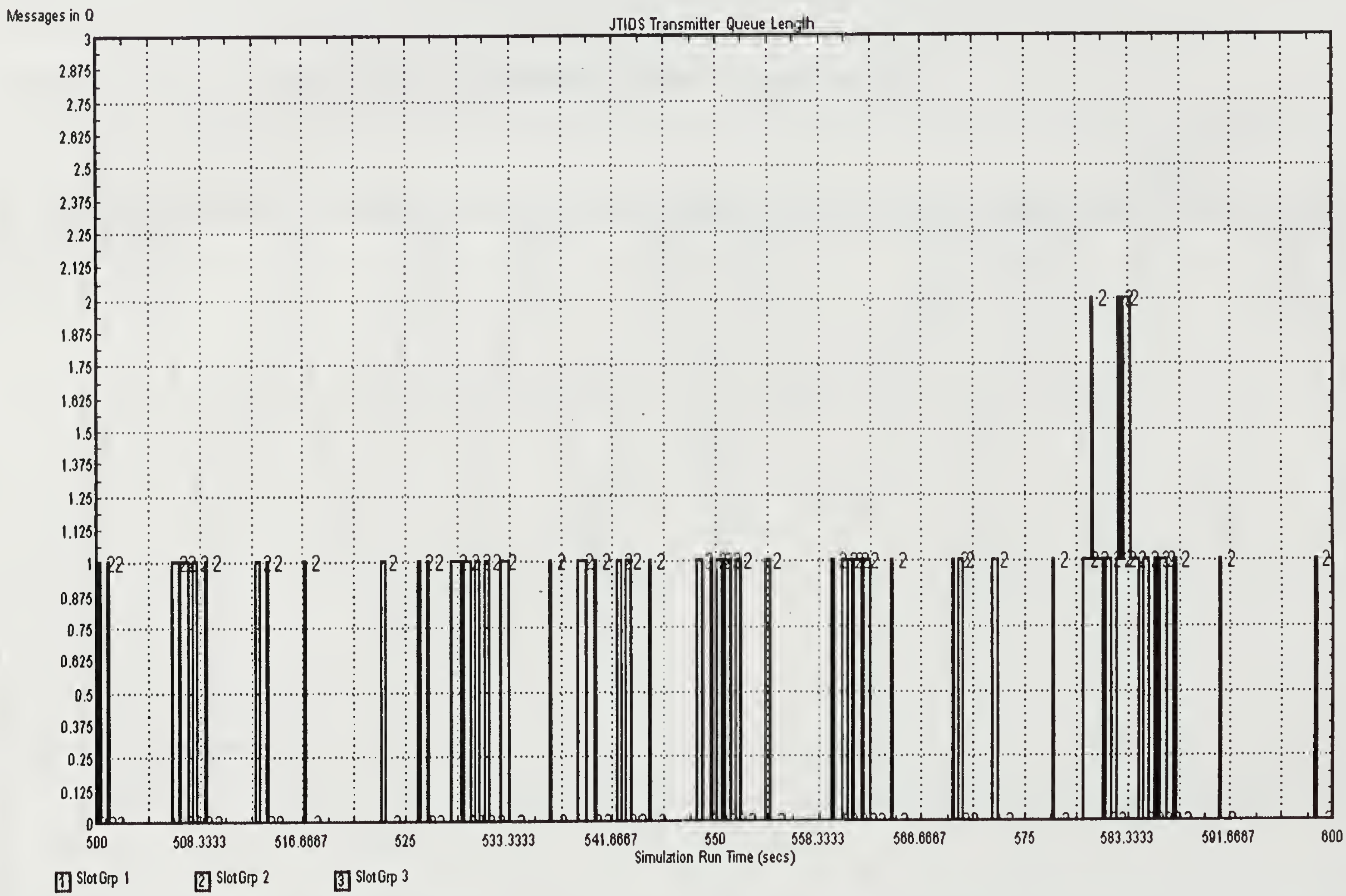

Figure 6-15. JTIDS Unit #3 Message Queue, EXTEND Model.

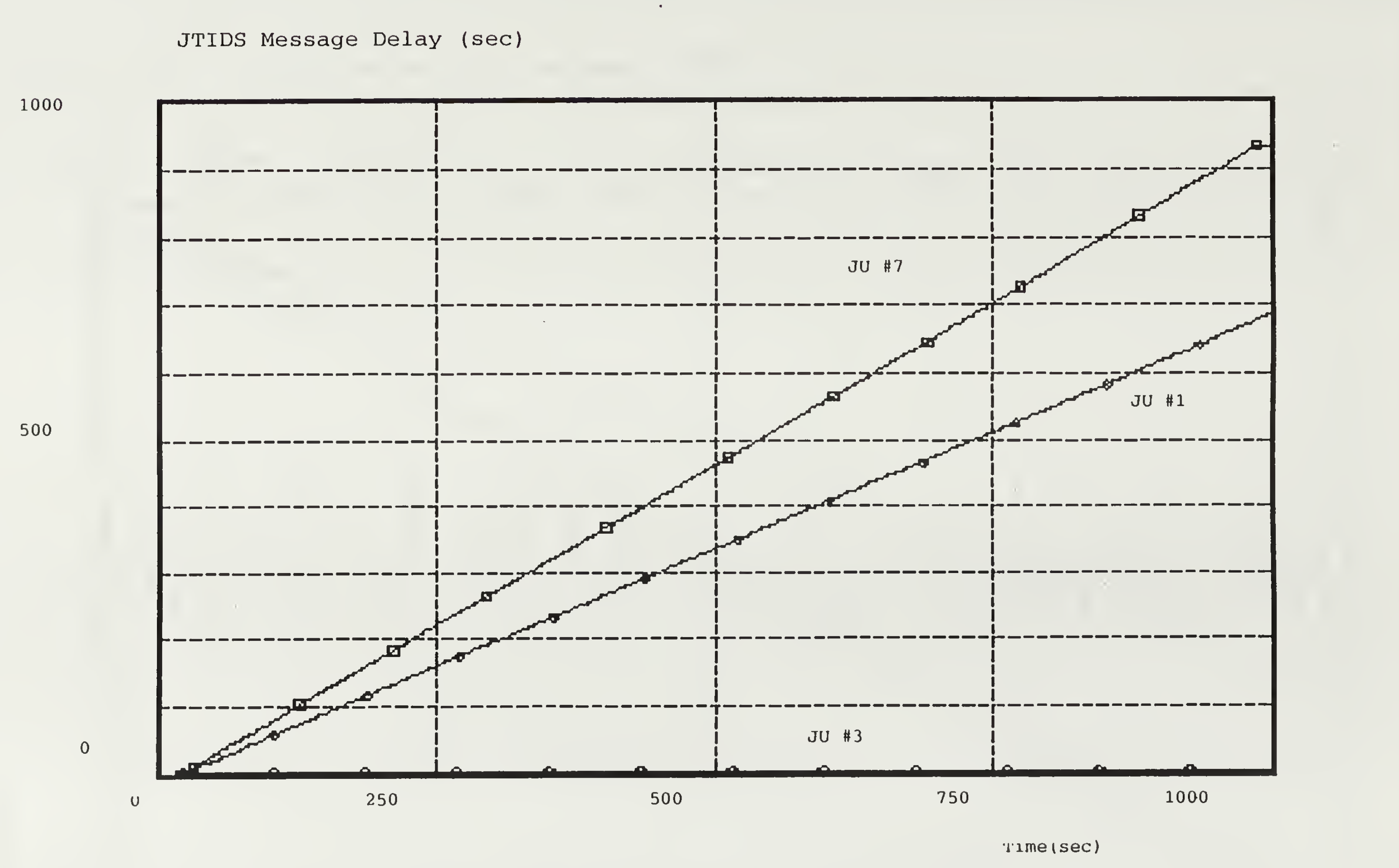

Figure 6-16. JTIDS Delay Time, OPNET Model.

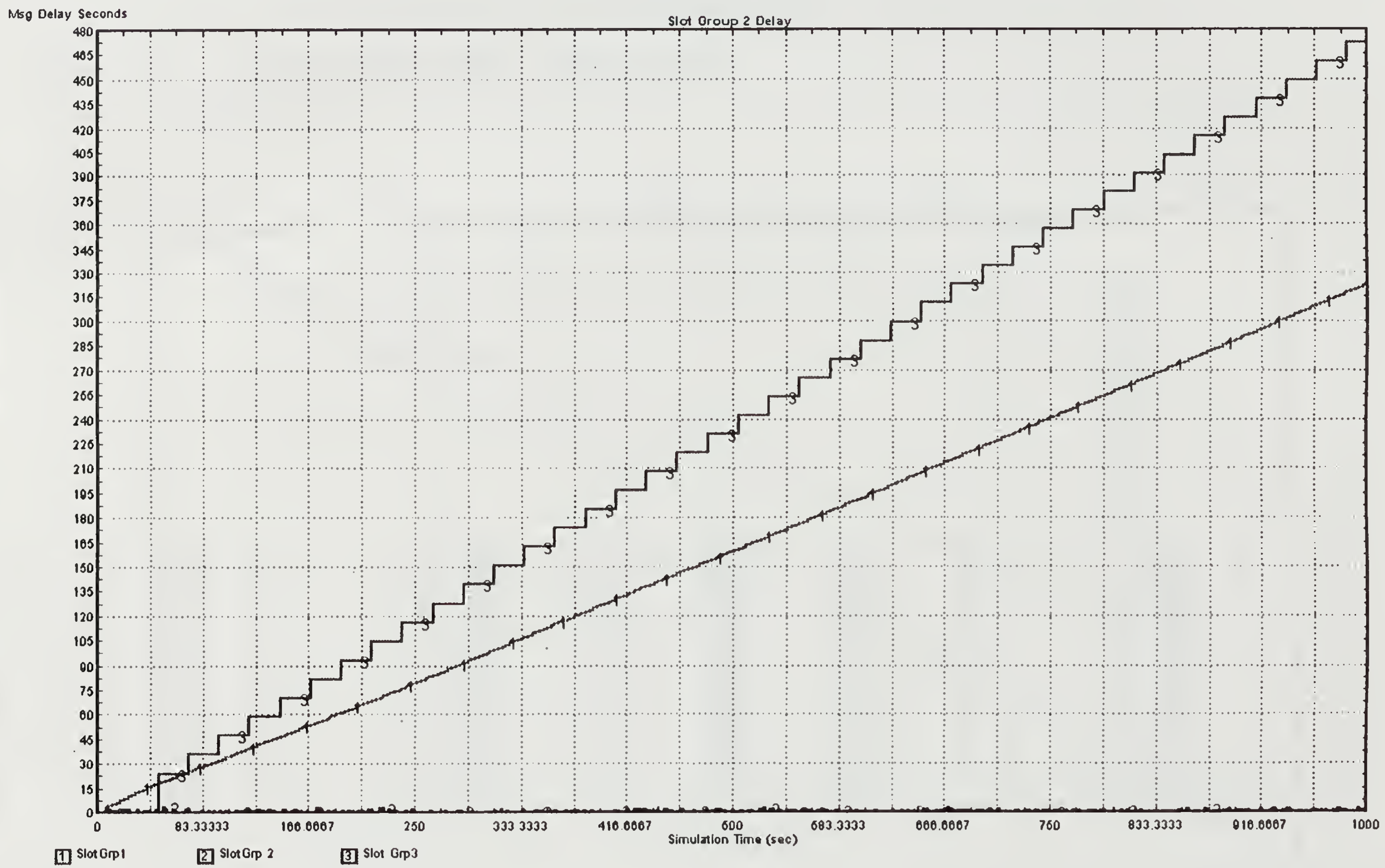

Figure 6-17. JTIDS Delay Time, EXTEND Model.

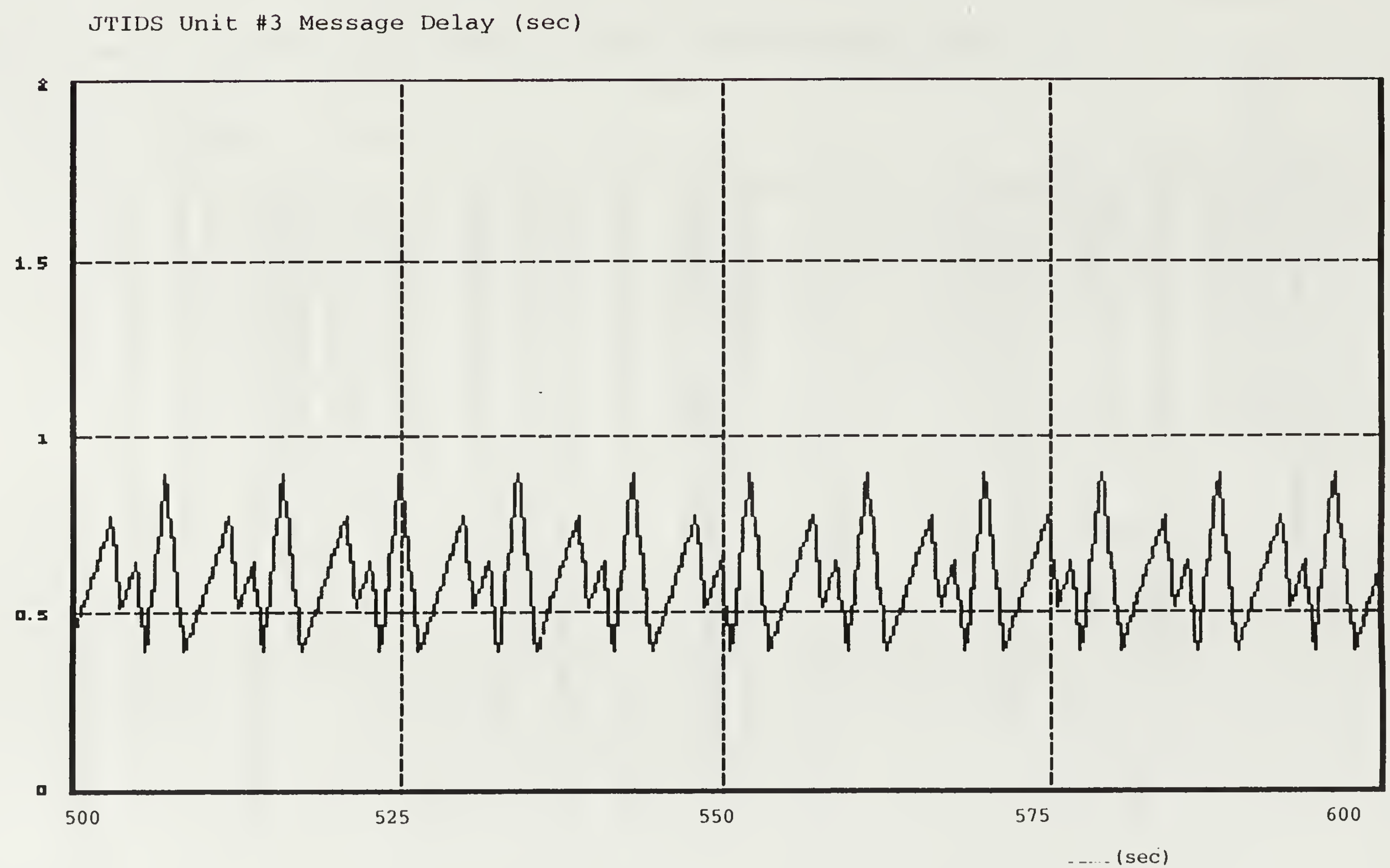

Figure 6-18. JTIDS Unit #3 Delay Time, OPNET Model.

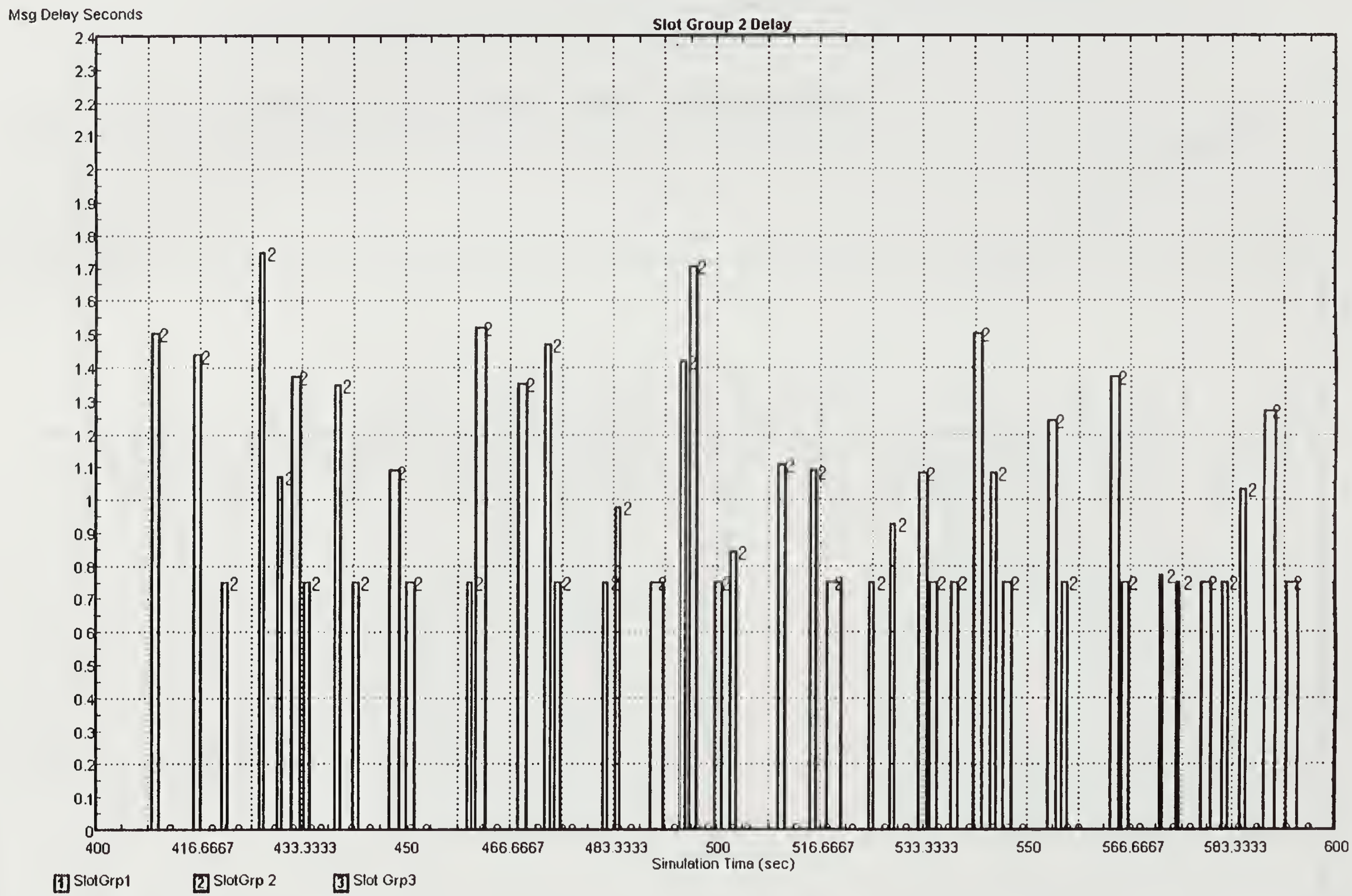

Figure 6-19. JTIDS Slot Group #2 Delay Time, EXTEND Model.

VII. CONCLUSION

A. SUMMARY

The overarching purpose of this research is provide unified command and joint task force communication planners with the best tools for planning and managing the increasing communications demand. Two goals were established to accomplish this. The first goal is to compare the performance of two computer-aided modeling and simulation tools. The second goal is to provide a subjective evaluation by using these modeling tools in an operational situation. Four computer models were developed, to simulate two very different communication architectures, using OPNET MODELER/RADIO by MIL3, and EXTEND by Imagine That. These goals were achieved through the modeling efforts and the simulation results obtained with the models.

B. THE TOOLS

The network models developed using OPNET and EXTEND produced very similar and believable results. There were

some significant differences that can be attributed to model design and not the tools. The system responses and trends were consistent between the two models even when the magnitude of the recorded performance measure differed. Most of the differences occurred between the two IT-21 models. These models were based on a heterogeneous ATM and Ethernet LAN subnet, linked to a second subnet via an ATM-based WAN. The discrepancies are attributed to differences in the cross network data load and the placement of system probes. In a future effort, these differences should be corrected to bring the two model results more in line.

C. APPLICATIONS

Perhaps more important than the numerical results are the lessons learned. The differences between the OPNET and EXTEND IT-21 models highlight the complexity of OPNET and the importance of understanding exactly what the tool is modeling. OPNET is a very powerful tool. With moderate time and training the packaged OPNET modules can be used to develop **network** models with some proficiency. To customize **process** modules to emulate new or unique systems requires a

"step increase" in the amount of resources (personnel training, experience, and time) to master.

At the other end of the complexity spectrum is EXTEND. This is also a very powerful tool, composed of very basic building blocks. The functionality of each of the EXTEND building blocks is very easy to understand. The blocks "stand alone" and to perform their designated function. Subroutines or function calls are all transparent to the user. These qualities make EXTEND easier to understand and more user friendly then OPNET. Based on this project, there is a much steeper learning curve with EXTEND, which means less training time to develop a working level-of-knowledge. These attributes are well suited for a military environment where tour lengths are two to three years. The simplicity of the blocks mandates that the modeler understands the systems higher level processes of the system being modeled. This makes EXTEND ideal for modeling the "big picture." For example, an EXTEND model representing the primary nodes and links in a network could be used as a "living" status board. When a capability is lost, gained, or proposed, remove or add the block corresponding to the object or capability on the "status board" model. Then examine system performance with the model to verify performance. If necessary, ship the electronic "board" to rear area experts for maintenance,

report the results of a site survey, clarify in-area communications, or resolve a problem. The simplicity, costs, and resources (man and machine) associated with EXTEND make the tool very portable, so modeling efforts can be distributed for larger projects. Custom-built objects can be placed in a public library and shared with others.

Another possible use is evaluating the communication architectures of field exercises. The particular exercise network architecture is entered into the model data base. Nodes generating traffic consistent with the operation represent the users. Once the network is populated with all the loads and sensors are in place, it can be "run" to find out where the weak links are located. It also can be used to "what if" system design on a broad level, capturing all the "back-of-the-envelope" calculations that experienced operators make routinely.

In closing, OPNET and EXTEND are only two of multitudinous COTS tools available. This study shows that a generic discrete-event modeling tool, such as EXTEND, can replicate, at lower levels, the results obtained with a more expensive network and communication modeling tool.

D. RECOMMENDED FUTURE STUDIES

Four areas for related research became apparent while working on this project. They are model verification, model abstraction, distributed model development, and modeling the communication network supporting a military operation.

1. Model Verification

Models are more credible if verified with actual networks. In this study, the OPNET model was considered the verified source. The results of the OPNET models were used as the benchmark for comparison of various modeling tools. The next step, beyond this thesis effort, is to collect traffic and application source information from the modeled network. The data collected, such as frame size, destinations, ETE delay times, throughput and peak loading, is then used as source information to derive the simulated network load and compare the model results with actual performance.

2. Model Abstraction

A second area for future research involves model abstraction. The IT-21 models developed for this project were high level, modeling the flow of individual cells or packets from each source in the network. Modeling individual cell flow from generation to destination required

a tremendous amount of computing resources, including the time to run the simulations. In follow-on efforts, using abstractions, groups of cells could be modeled instead of individual cells. Multiple users or an entire LAN might be represented as a unit, based on results of a few high-level models. The trade-off it be determined is how much can be abstracted and still obtain a "good enough" solution.

3. Distributed Model Development

Another area for study concerns distributed model development, using an object-oriented approach. Two characteristics of modeling became very obvious during this project. First, modeling can be very time-consuming and labor-intensive. Second, a thorough understanding, of the system to be modeled, is critical to developing reliable models. The question to be answered is "using techniques similar to software development, is it feasible to distribute the development of model objects, which make up larger networks, and successfully integrate them?" This would be useful when modeling heterogeneous systems or supporting a small, forward element such as an advanced planning team. Calling on the proper resources to contribute model development would distribute the workload and expertise.

4. Operational Network Models

Finally, future studies could take this project to the next level by developing a model of the communication network associated with a military operation and evaluating the level of effort involved. For example, model the radio frequency links (satellite, line-of-sight, high frequency) and directed communications associated with a small operation such as a special operations team or a non-combatant operation. Keep the model at a broad level but with the fidelity to track interoperability between sources, data rates, and system performance. Experimental systems or new combinations could be considered. For example, using Global Broadcast System (GBS) for a large bandwidth feed to a remote user that has a low-bandwidth, demand-assigned multiple-access (DAMA) unit to reach back to the GBS station.

There are many possible applications for modeling and simulation. If modeling and simulation is going to benefit the warfighter, then the tools or the products need to get to the operators in a useful form. Identifying the capabilities and limitations of modeling and simulation tools, as they apply to command and control networks, is a step in that direction.

APPENDIX A. OPNET NETWORK REPORT, JTIDS MODEL

...
...

Keywords			
attribute	*value*	*type*	*default value*
keywords	custom_model_list	typed file	
	general	typed file	

subnet **subnet_0**			
attribute	*value*	*type*	*default value*
name	subnet_0	string	n
priority	0	integer	0
user id	0	integer	0
x position	-97.3638050682261	double	0.0
y position	27.8088265107213	double	0.0
x center	-97.3638050682261	double	0.0
y center	27.8088265107213	double	0.0
x span	0.402861598440546	double	0.0
y span	0.322666666666667	double	0.0
threshold	0.0	double	0.0
map	usa	typed file	NONE
icon name	subnet	icon	subnet
outline color	RGB133	color	RGB133
mobile_0.node_flag	disabled	toggle	disabled
mobile_0.JTIDS.Node Set	2	integer	0
....JTIDS.Rate Redundancy Number	9.0	double	0.0
mobile_0.JTIDS.Start Slot	4	integer	0
mobile_0.run_ctr	promoted	double	0.0
mobile_1.node_flag	disabled	toggle	disabled
mobile_1.JTIDS.Node Set	2	integer	0
....JTIDS.Rate Redundancy Number	9.0	double	0.0
mobile_1.JTIDS.Start Slot	36	integer	0
mobile_2.node_flag	disabled	toggle	disabled
mobile_2.JTIDS.Node Set	1	integer	0
....JTIDS.Rate Redundancy Number	11	double	0.0
mobile_2.JTIDS.Start Slot	3	integer	0
mobile_3.node_flag	disabled	toggle	disabled
mobile_3.JTIDS.Node Set	1	integer	0
....JTIDS.Rate Redundancy Number	11	double	0.0
mobile_3.JTIDS.Start Slot	11	integer	0
mobile_4.node_flag	disabled	toggle	disabled
mobile_4.JTIDS.Node Set	2	integer	0
....JTIDS.Rate Redundancy Number	9.0	double	0.0
mobile_4.JTIDS.Start Slot	20	integer	0
mobile_5.node_flag	disabled	toggle	disabled
mobile_5.JTIDS.Node Set	2	integer	0
....JTIDS.Rate Redundancy Number	9.0	double	0.0
mobile_5.JTIDS.Start Slot	52	integer	0
mobile_6.node_flag	disabled	toggle	disabled
mobile_6.JTIDS.Node Set	2	integer	0
....JTIDS.Rate Redundancy Number	6.0	double	0.0
mobile_6.JTIDS.Start Slot	18	integer	0
mobile_7.node_flag	disabled	toggle	disabled
mobile_7.JTIDS.Node Set	2	integer	0

....JTIDS.Rate Redundancy Number	6.0	double	0.0
mobile_7.JTIDS.Start Slot	274	integer	0

fixed node subnet_0.mobile_0

attribute	*value*	*type*	*default value*
name	mobile_0	string	f
model	my_link16_rt	enumerated	NONE
x position	6.34137595406159	double	0.0
y position	11.0873682576332	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0
run_ctr	promoted	double	0.0

mobile node subnet_0.mobile_1

attribute	*value*	*type*	*default value*
name	mobile_1	string	m
model	my_link16_rt	enumerated	NONE
x position	5.99766770418132	double	0.0
y position	7.09366047458937	double	0.0
trajectory	NONE	typed file	NONE
color	RGB030	color	RGB030
threshold	0.0	double	0.0
icon name	mobile_comm	icon	mobile_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

fixed node subnet_0.mobile_2

attribute	*value*	*type*	*default value*
name	mobile_2	string	f
model	my_link16_rt	enumerated	NONE
x position	10.3579809895381	double	0.0

...
...

y position	10.4775433141409	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

fixed node subnet_0.mobile_3

attribute	*value*	*type*	*default value*
name	mobile_3	string	f
model	my_link16_rt	enumerated	NONE
x position	11.8589409048735	double	0.0
y position	5.40764350022129	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

fixed node subnet_0.mobile_4

attribute	*value*	*type*	*default value*
name	mobile_4	string	f
model	my_link16_rt	enumerated	NONE
x position	12.2533462023483	double	0.0
y position	16.2933876538685	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

...
...

fixed node subnet_0.mobile_5			
attribute	*value*	*type*	*default value*
name	mobile_5	string	f
model	my_link16_rt	enumerated	NONE
x position	15.755370272974	double	0.0
y position	5.94968464216504	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

fixed node subnet_0.mobile_6			
attribute	*value*	*type*	*default value*
name	mobile_6	string	f
model	my_link16_rt	enumerated	NONE
x position	16.2547511782956	double	0.0
y position	18.6350998185092	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled
node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

fixed node subnet_0.mobile_7			
attribute	*value*	*type*	*default value*
name	mobile_7	string	f
model	my_link16_rt	enumerated	NONE
x position	21.1888210936766	double	0.0
y position	9.69120319328274	double	0.0
threshold	0.0	double	0.0
icon name	fixed_comm	icon	fixed_comm
altitude	15	double	0.0
condition	enabled	toggle	enabled

...
...

node_flag	promoted	toggle	disabled
priority	0	integer	0
user id	0	integer	0
JTIDS.Node Set	promoted	integer	0
JTIDS.Rate Redundancy Number	promoted	double	0.0
JTIDS.Start Slot	promoted	integer	0

APPENDIX B. IT-21 EXTEND MODEL PARAMETERS

Ethernet Settings

This is the MTU, in bytes, for the ETHERNET (E-mail), Default is 1500Bytes.	1500
This is the rate that the edge device converts ETHNET E-mail Packets to ATM Cells in ETHNET Pkts/sec. Default should be 8300 pkts/sec.	8300
This is the rate that the edge device converts ETHNET FTP Packets to ATM Cells in ETHNET Pkts/sec. Default should be 8300 pkts/sec.	8300

Ethernet Workstation Settings

ETH-Mail WS 1

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ETH-Mail WS 2

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ETH-Mail WS 3

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ETH-Mail WS 4

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ETH-Mail WS 5

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ETH-Mail WS 6

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ETH-ATM WS Settings

ETH-FTP WS 1

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	5000

ETH-FTP WS 2

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	5000

ETH-FTP WS 3

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	5000

ETH-FTP WS 4

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	5000

ETH-FTP WS 5

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	5000

ETH-FTP WS 6

User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	5000

ATM Settings

ATM Mail WS

Select ATM Quality of Service. Set to (-1) for QoS A (VTC) and set to (0) for QoS D (All other Data)	0
User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	7200
Message Size in Bytes (normal pdf)	2000
Std Deviaion	200

ATM FTP WS

Select ATM Quality of Service. Set to (-1) for QoS A (VTC) and set to (0) for QoS D (All other Data)	0
User Selected Msg Rate (msgs/hr). Poisson Deliverate (exponential arrival interval	3600
Message Size in Bytes (normal pdf)	50000
Std Deviaion	500

ATM VTC Settings

Select VTC Quality of Service. Set to (-1) for QoS A (VTC) and set to (0) for QoS D (All other Data)	-1
VTC Conference Duration (minutes).	4.0
VTC Conference Interval (Conferences/Day) >= 1.	7200
VTC Frame Size (bytes/frame) Default 100000 bytes/frame.	100000
VTC Frame Rate (frames/sec) Default 20 frames/sec.	30

APPENDIX C. JTIDS EXTEND MODEL PARAMETERS

JTIDS Unit #1

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	9
Index Number (integer btwn 0-511)	4
SET (Set A = 1, Set B = 2, Set C =3)	3

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 1
- ○ Weibull

Mean= 1

Std Dev= 0.5

1

Message Size for Fixed Format J-Series Messages: 3

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize):

Max Message size (< 560 bits):

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = 360
- ○ Weibull

Min = 359

Max = 361

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform — Min= 100000
- ○ LogNormal — Max= 100001
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= ######
- ○ Weibull

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize):

Max Message size (< 3600 bits):

- ○ Binomial
- ○ Erlang
- ○ Exponential — Min = 0
- ○ HyperExponential — Max = 1
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = 1400
- ○ Weibull

JTIDS Unit #2

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	9
Index Number (integer btwn 0-511)	36
SET (Set A = 1, Set B = 2, Set C =3)	3

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 15
- ○ Weibull

Mean= 1

Std Dev= 0.5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal

Min = 358

Max = 360

○ Poisson
○ Real, uniform
○ Triangular: Most Likely = [360]
○ Weibull

Message Generation Rate for Free Text Msgs (seconds)

○ Binomial
○ Constant
○ Erlang
○ Exponential
○ HyperExponential
◉ Integer, uniform
○ LogNormal
○ Normal
○ Poisson
○ Real, uniform
○ Triangular: most likely value= [ ]
○ Weibull

Min= [10000]
Max= [10001]

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

○ Binomial
○ Exponential
◉ Integer, uniform
○ LogNormal
○ Normal
○ Poisson
○ Real, uniform
○ Triangular: Most Likely =

Min = [0]
Max = [1]

JTIDS Unit #3

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	11
Index Number (integer btwn 0-511)	3
SET (Set A = 1, Set B = 2, Set C =3)	2

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 1
- ○ Weibull

Mean= 1

Std Dev= 0.5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ◉ Triangular: Most Likely = 360

Min = 358

Max = 360

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= [ ]
- ○ Weibull

Mean=	36
Std Dev=	6

Message Size for Free Text Messages: **Mean =** 2400

Minimum Message size (>35 bits): **Std Dev =** 400

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

○ Binomial	◉ Normal
○ Exponential	○ Poisson
○ Integer, uniform	○ Real, uniform
○ LogNormal	○ Triangular: Most Likely = [ ]

JTIDS Unit #4

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	11
Index Number (integer btwn 0-511)	11
SET (Set A = 1, Set B = 2, Set C =3)	2

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ **Binomial**
- ○ **Constant**
- ○ **Erlang**
- ○ **Exponential**
- ○ **HyperExponential**
- ○ **Integer, uniform**
- ○ **LogNormal**
- ◉ **Normal**
- ○ **Poisson**
- ○ **Real, uniform**
- ○ **Triangular: most likely value=** 1
- ○ **Weibull**

Mean= 1

Std Dev= 0.5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

Min = 358

Max = 360

- ○ **Binomial**
- ○ **Erlang**
- ○ **Exponential**
- ○ **HyperExponential**
- ◉ **Integer, uniform**
- ○ **LogNormal**
- ○ **Normal**
- ○ **Poisson**
- ○ **Real, uniform**
- ○ **Triangular: Most Likely =** 360
- ○ **Weibull**

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= [ ]
- ○ Weibull

Min= 10000

Max= 10001

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

Min = 0

Max = 1

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = [ ]

JTIDS Unit #5

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	9
Index Number (integer btwn 0-511)	20
SET (Set A = 1, Set B = 2, Set C =3)	3

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 1
- ○ Weibull

Mean= 1

Std Dev= 0.5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

Min = 358

Max = 360

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = 2e+005

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= [ ]
- ○ Weibull

Min=	10000
Max=	10001

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

Min =	0
Max =	1

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = [ ]

JTIDS Unit #6

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	9
Index Number (integer btwn 0-511)	52
SET (Set A = 1, Set B = 2, Set C =3)	3

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 1
- ○ Weibull

Mean= 1

Std Dev= 0.5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

Min = 358

Max = 360

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = 360

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= [ ]

Min=	10000
Max=	10001

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

Min =	0
Max =	1

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = [ ]

JTIDS Unit #7

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	6
Index Number (integer btwn 0-511)	18
SET (Set A = 1, Set B = 2, Set C =3)	3

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 1
- ○ Weibull

Mean= 1

Std Dev= .5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

Min = 358

Max = 360

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ◉ Triangular: Most Likely = 360

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= [ ]
- ○ Weibull

Min= 10000

Max= 10001

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

Min = 0

Max = 1

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = [ ]

JTIDS Unit #8

Set JTIDS Parameters:

Data Packing Structure: Std Pack = 1; P2 = 2	1
Recurrance Rate Number (RRN) (Integer btwn 0 and 15)	6
Index Number (integer btwn 0-511)	274
SET (Set A = 1, Set B = 2, Set C =3)	3

Message Generator Parameters:

Message Generation Rate for Fixed Format J-Series Msgs (seconds):

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ○ Integer, uniform
- ○ LogNormal
- ◉ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= 1
- ○ Weibull

Mean= 1

Std Dev= .5

Message Size for Fixed Format J-Series Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 560 bits):

Min = 358

Max = 360

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = 360

Message Generation Rate for Free Text Msgs (seconds)

- ○ Binomial
- ○ Constant
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: most likely value= [ ]
- ○ Weibull

Min= 10000

Max= 10001

Message Size for Free Text Messages:

Minimum Message size (>35 bits):

Most Likely Size (MinSize<MostLikely<MaxSize)

Max Message size (< 3600 bits):

Min = 0

Max = 1

- ○ Binomial
- ○ Erlang
- ○ Exponential
- ○ HyperExponential
- ◉ Integer, uniform
- ○ LogNormal
- ○ Normal
- ○ Poisson
- ○ Real, uniform
- ○ Triangular: Most Likely = [ ]

APPENDIX D. GLOSSARY OF TERMS

3-D	Three-dimensional
ABR	Available Bit Rate
AMS	ATM Model Suite
ATM	Asynchronous Transfer Mode
ATM	Asynchronous Transfer Mode
BEES	Battle Force EMI Evaluation System
BER	Bit Error Rate
B-ISDN	Broadband Integrated Services Digital Network
BPS	Bits per Second
C2	Command and Control
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CASE	Computer Software Engineering
CBR	Constant Bit Rate
CCSK	Cyclic Code Shift Keying
CLP	Cell Loss Priority
CLP	Cell Loss Priority
CONUS	Continental United States
COTS	Commercial off-the-shelf
CPSM	Continuous Phase Shift Modulation

DAMA	Demand-Assigned Multiple-Access
DLL	Dynamic-Link Libraries
DS/DS	Desert Storm/Desert Shield
EME	Electromagnetic Environment
EMI	Electromagnetic Interference
EPLRS	Enhance Position Location Reporting System
ETE	End-to-End
EW	Electronic Warfare
FIFO	First-In-First-Out
FTP	File Transfer Protocol
GBS	Global Broadcast System
GFC	Generic Flow Control
GIE	Global Information Environment
GUI	Graphical User Interface
HEC	Header Error Control
IEEE	Institute for Electrical and Electronic Engineers
IFF	Identify Friend or Foe
IP	Internet Protocol
IT-21	Information Technology for the 21st Century
ITU	International Telecommunications Union
ITU-T	ITU Telecommunications Standardization Sector
JFC	Joint Force Commander
JTF	Joint Task Force
JTIDS	Joint Tactical Information Distribution System

JU	JTIDS Unit
KBPS	Kilobits per Second
LAN	Local Area Network
LILO	Last-In-First-Out
LOS	Line-of-sight
M3UI	MIL 3 User Inter-face
MBPS	Megabits per Second
MIL3	Modeling Technologies for the Third Millennium
NATO	North American Treaty Organization
NPG	Network Participation Group
OAM	Operations, Administration and Maintenance
OOTW	Operations other than war
OPNET	Optimized Network Engineering Tools
P2DP	Packed-2 Double Pulse
PBX	Private Branch Exchange
PC	Personal Computer
PTI	Payload Type Identifier
QoS	Quality of Service
RAM	Random Access Memory
RRN	Recurrence Rate Number
RTT	Round-trip Timing
SA	Situational Awareness
SABER	Situational Awareness Beacon with Reply
SAR	Segmentation and Reassembly

SIPRNET	Secure IP Router Network
STD-DP	Standard-Double Pulse
TADIL	Tactical Digital Information Link
TDMA	Time Division Multiple Access
UBR	Unspecified Bit Rate
UN	United Nations
UNI	User-network Interface
US	United States
VBR	Variable Bit Rate
VCI	Virtual Channel Identifier
VPI	Virtual Path Identifier
VTC	Video Teleconference
WAN	Wide Area Network

APPENDIX E. POINTS OF CONTACT

Bond, Hank, LCDR, USN
Office of Warfighter Support
Joint Interoperability Test Command
Ft Huachuca, AZ 85613
DSN: 879-4328
E-mail: bonh@fhu.disa.mil

Cole, Gary H., LTC, USAF
Director of Operations
DoD Joint Spectrum Center
Anapolis, MD 21402-5064
DSN: 281-9823
E-mail: cole@jsc.mil

LaGaspe, Albert, PhD
Modeling Advanced Concepts, PMW-131
SPAWAR Systems Center
San Diego, CA 92152
DSN: 577-0180
E-mail: legaspi@spawar.navy.mil

LIST OF REFERENCES

1. William Stallings, Data and Computer Communications, 5th Edition, 1997 Prentice-Hall Inc., Upper Saddle River, New Jersey.

2. Joint Pub 6-0, Doctrine for Command, Control, Communications, and Computers (C4), Systems Support to Joint Operations, 30 May 1995.

3. Joint Pub 1, Joint Warfare of the Armed Forces of the United States, 10 January 1995.

4. Concept for Future Joint Operations (CFJO), Expanding Joint Vision 2010, May 1997, Chairman of the Joint Chiefs of Staff, Washington, D.C.

5. Joint Vision 2010, Chairman of the Joint Chiefs of Staff, Washington, D.C.

6. CNO N6 Copernicus Home Page, Department of the Navy, http://diamond.spawar.navy.mil/specs/wise/wiseup/copern.html/, 1995.

7. EXTEND Users Manual, Imagine That Inc, 1995.

8. Sullivan, J. E., A Computer Network Simulation Tutorial for COMNET III, Masters Thesis, Naval Postgraduate School, Monterey, CA, June 1996.

9. Imagine That Inc., http://www.imaginethatinc.com/, 1998.

10. Evaluation of Automated Tools to Support Physical Modeling, AFC4A Technical Report: TN TR 95-07, Air Force C4 Agency Directorate of Interoperability and Technology (HQ AFC4A/TN), 8 March 1995.

11. Battle Force EMI Evaluation System (Bees) System Description, SPAWAR Systems Center, September 1997.

12. CACI Corporation, COMNET III Users Manual, 1993.

13. Altmann, M., Computer Simulation Software Frequently Asked Questions, Version 3.0, http://www.nmsr.labmed.umn.edu/~michael/dbase/comp-simulation.html/, 12 Jan 1996.

14. Integrated Systems Inc., Column 0032-9/97, 1997.

15. Reilly, P. L., SimTools Version 2.0, Delphi Internet Services Corporation, July 1994.

16. OPNET Modeler Tutorial Manual, MIL3 Inc., 1997.

17. Abstraction Software Network Simulation Modeling Workflow Prophesy, http://www.sni.net/abstraction/, 1998.

18. Scientific and Engineering Software Inc., http://www.ses.com/wb_NT.html/, January 9, 1998.

19. Scientific and Engineering Software Inc., http://www.ses.com/strategizer.html/, December 9, 1997.

20. SPECTRUM XXI Homepage, http://www.jsc.mil.specxxi/, 1998.

21. Vertal, M. D., System Modeling and Simulation with Foresight, Computing for Engineers, Issue 19, http://wwwcn.cern.ch/ce/newsletter/issue19/ cse19_foresight.html/, February 1996.

22. Ricciardi, S., Diagram Your Network with NetViz 2.5, Computer Shopper, http://www.zdnet.com/cshopper/content /9608/cshp0147.html/, August 1996.

23. Byrd, Valerie R, The Integration of Situational Awareness Beacon with Reply (SABER) with the Enhanced Position Location Reporting System (EPLRS), Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1996.

24. Understanding Link-16, Navy Center for Tactical Systems Interoperability, San Diego, CA, April 1994.

25. OPNET Model Library: ATM, http://www.mil3.com/products/library /atm.html, 1998.

26. Whitaker, J., Transition to Digital: The Ins and Outs of ATM., Broadcast Engineering, July 1, 1997.

27. Legaspi,. A., PhD, Joint Maritime Systems Analysis Center (JMSAC) Brief, Version 17, SPAWAR PMW-131-41, February 9, 1998.

28. OPNET Modeler Manual, Volume 1, MIL3, Inc., 1997.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Professor William Kemple, CC/KE 1
   Naval Postgraduate School
   Monterey, CA 93943-5101

4. Professor John Osmundson, CC/OS 1
   Naval Postgraduate School
   Monterey, CA 93943-5101

5. Lieutenant Commander Alan R. Rieffer 1
   P.O. Box 992
   Florence, AZ 93940